ARMSYS2410 开发套件

Linux 用户手册

（Ver 1.3）

杭州立宇泰电子有限公司

http://www.hzlitai.com.cn

目 录

1．内容综述....4
2．硬件环境....4
2．1 串口的使用....5
2．2 JTAG 的使用....7
2．3 网络通讯的使用....7
3．软件环境....8
3．1 安装交叉编译器....8
3．2 准备好 Linux 内核源码包....8
3．3 准备好 root 文件系统....14
3．4 准备好 vivi Bootloader....14
4．VIVI BOOTLOADER 固化到 ARMSYS2410....15
5．KERNEL（LINUX 内核）固化到 ARMSYS2410....17
6．NFS 方式 MOUNT 主机....21
6．1 在主机上启动 nfs 服务....21
6．2 在目标板上进行设定....23
6．3 启动 Linux....24
7．NFS 方式更新文件系统....25
7．1 更新文件系统....25
7．2 从 YAFFS 文件系统启动....26
8．主机上开发应用程序....27
8．1 编写 Helloworld 代码....28
8．2 编译 hello....28
9．将应用程序下载到 ARMSYS2410....28
9．1 采用 NFS 方式....28
9．2 采用 U 盘拷贝....28
9．3 采用 FTP 方式....29
10．QT 环境下进行设计开发....29
10．1 基于 PC（主机）进行开发....29
10．2 建立基于 ARM 的开发环境....36
10．3 重建根文件系统....38

附录 A：内核驱动程序资源....39

A.1 串口驱动....39
A.2 MTD 驱动....39
A.3 ADC 驱动....39
A.4 4 按键驱动....39
A.5 RTC 驱动....39
A.6 触摸屏驱动....39
A.7 USB HOST 驱动....39
A.8 CS8900A 网卡驱动....39
A.9MMC 卡驱动....40
A.10 USB 摄像头驱动....40
A.11UDA1341 声卡驱动....40

A.12 LCD 驱动 40
A.12 LED 驱动 40
A.13 YAFFS 文件系统源码 40
A.14 其他资源 40

附录 B：实例程序 41

B.1 LED 跑马灯程序 41
B.2 LED 单独控制 41
B.3 按键控制 41
B.4 控制台下的 MP3 播放器 42
B.5 通过 USB 摄像头抓图 VIDCAT 42
B.6 WEB 服务器程序 42
B.7 QT 下的 HELLO 程序 42

附录 C：网络服务 43

C.1 IFCONFIG 命令 43
C.2 PING 命令 44
C.3 TRACEROUTE 命令 44
C.4 远程登陆 TELNET 44
C.5 TELNET 服务器 INETD 45
C.6 远程文件传送 FTP 45
C.7 WEB 服务器 BOA 45

1. 内容综述

本文将对以下内容进行详细说明，您会从中了解如何建立基于 ARMSYS2410 的嵌入式 Linux 开发环境、及如何在这个环境下进行开发工作。

（1） 建立硬件环境；

（2） 安装交叉编译工具，为目标板准备好 kernel 映像文件，root 文件系统，vivi Bootloader；

（3） 将 vivi Bootloader 固化到目标板；

（4） 将 kernel 映像文件固化到目标板；

（5） NFS 方式 mount 主机上的 root 文件系统；

（6） 通过 NFS 方式更新文件系统（支持 YAFFS 可读写的文件系统）；

（7） 在主机上为目标板开发应用程序；

（8） 将应用程序下载到目标板并运行；

（9） 安装和编译 Qt 开发环境；

（10） 开发基于 Qt 的应用程序，并发布到 ARMSYS2410；

（11） 附录 A：kernel 中提供的底层驱动和资源；

（12） 附录 B：ARMSYS2410 提供的实例程序资源；

（13） 附录 C：Linux 下提供的网络服务及应用程序资源；

本文档假设您已经在主机上安装了 Red Hat Linux9.0（完全安装方式）。

如果你的主机上安装了 Linux 其他发行版，可能会发生一些问题。因此如果您想避免麻烦并节省时间，最好在在您的 PC 上安装 Red Hat Linux9.0。

如果您想了解更多关于 Red Hat Linux 9.0 的安装问题，请查阅其他资料，或浏览：http://www.redhat.com/docs/manuals/linux/RHL-9-Manual/。

关于 ARMSYS2410 其它相关文档《ARMSYS2410 用户手册》、《ARMSYS2410 WinCE 用户手册 》。

2. 硬件环境

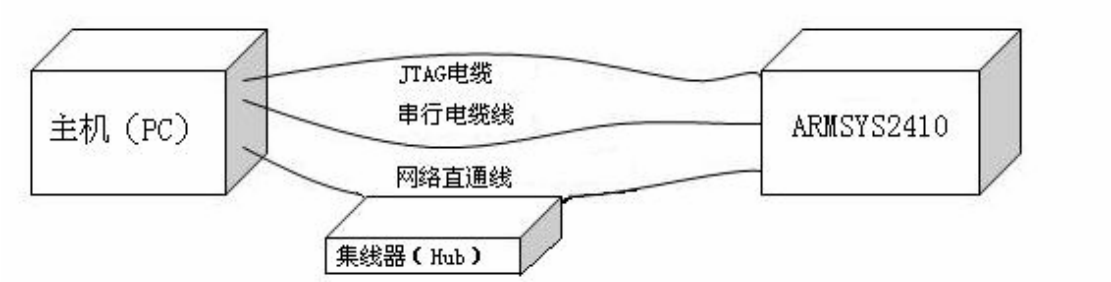


图 1 连接图（采用 hub 接入网络）

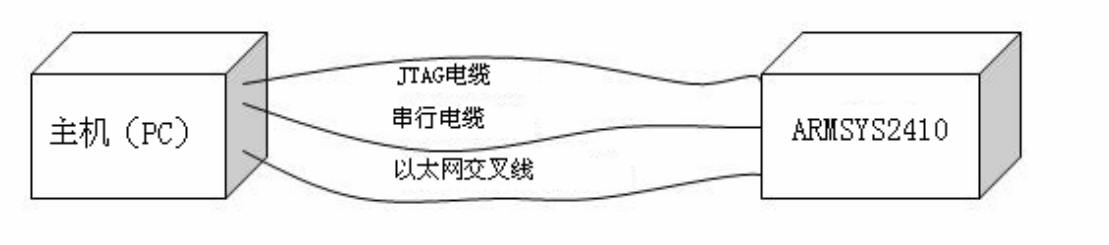


图 2 连接图（采用交叉对等线连接主机）

需要采用上图中的 3 根电缆线连接主机和开发板。下面分别介绍它们。

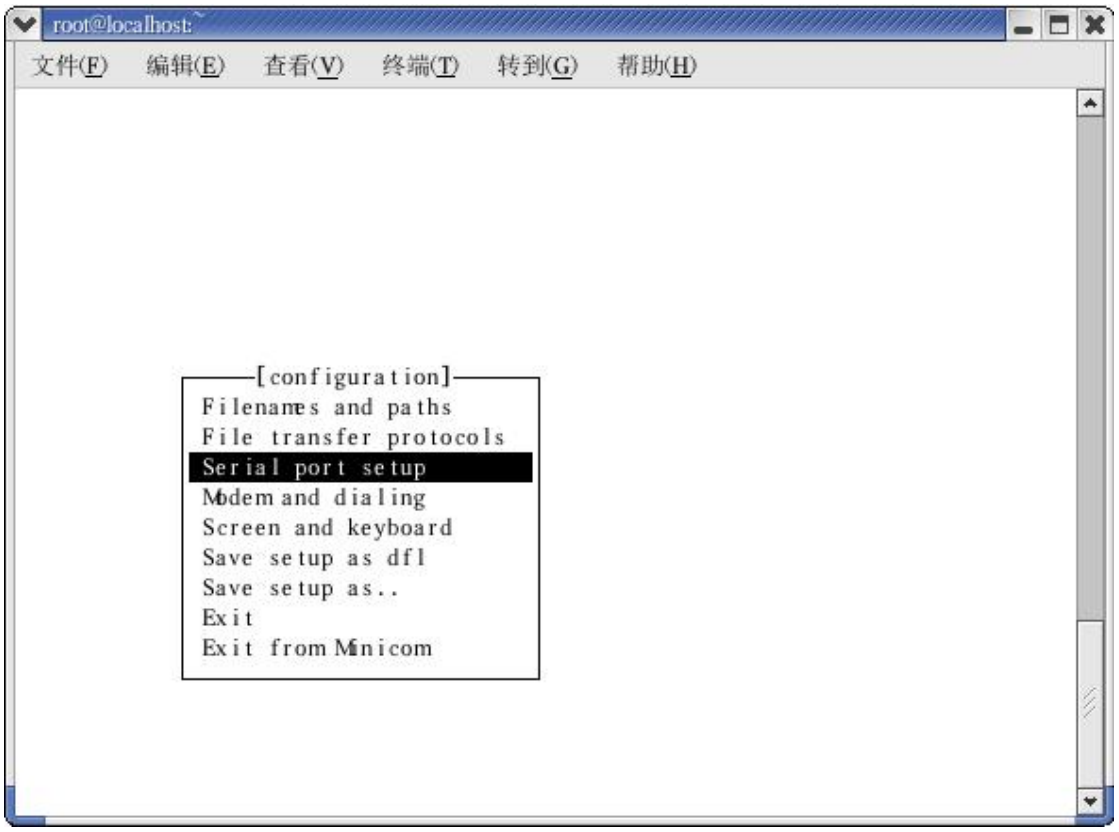
2．1 串口的使用

串口用来观察目标板在程序运行过程中的输出信息，同时也用来向目标板发送命令。采用 ARMSYS2410 配套的串口线连接好 PC 机的 COM 口与开发板的 COM1 口。在主机端，我们可以使用 minicom 来与 ARMSYS2410 的串口通讯。

配置 Minicom，输入命令行：

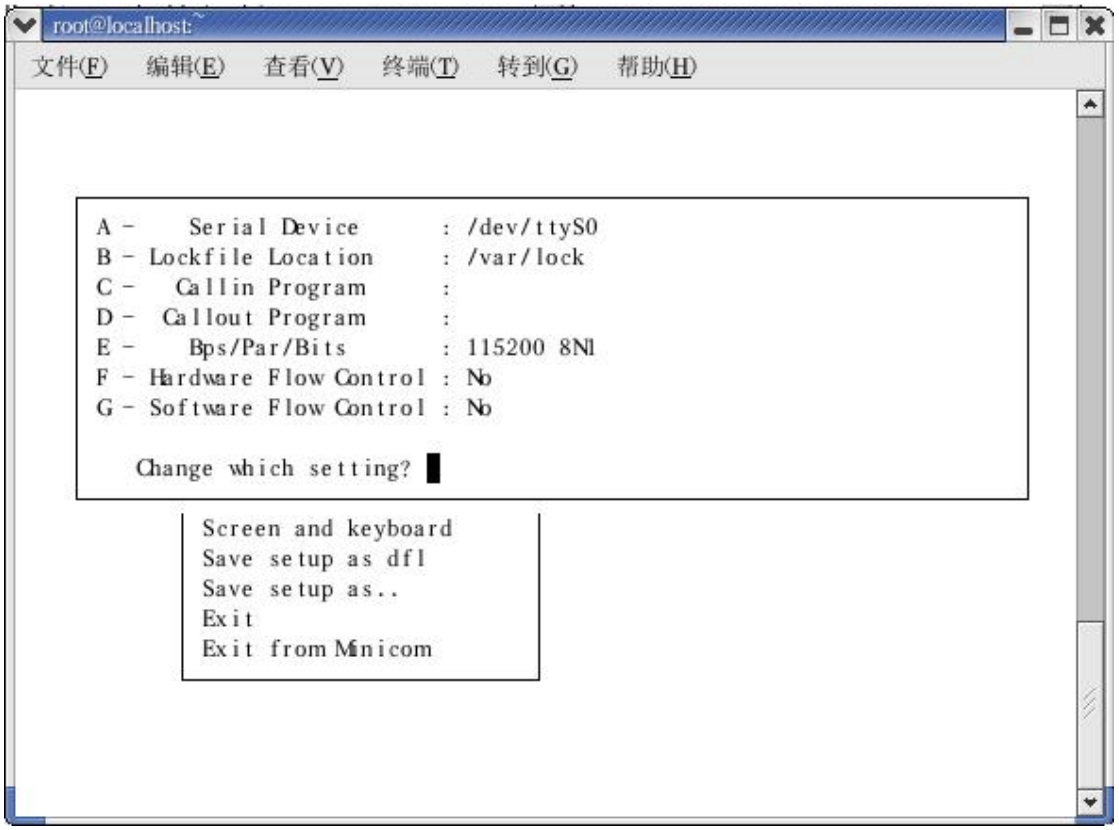
minicom –s

出现：



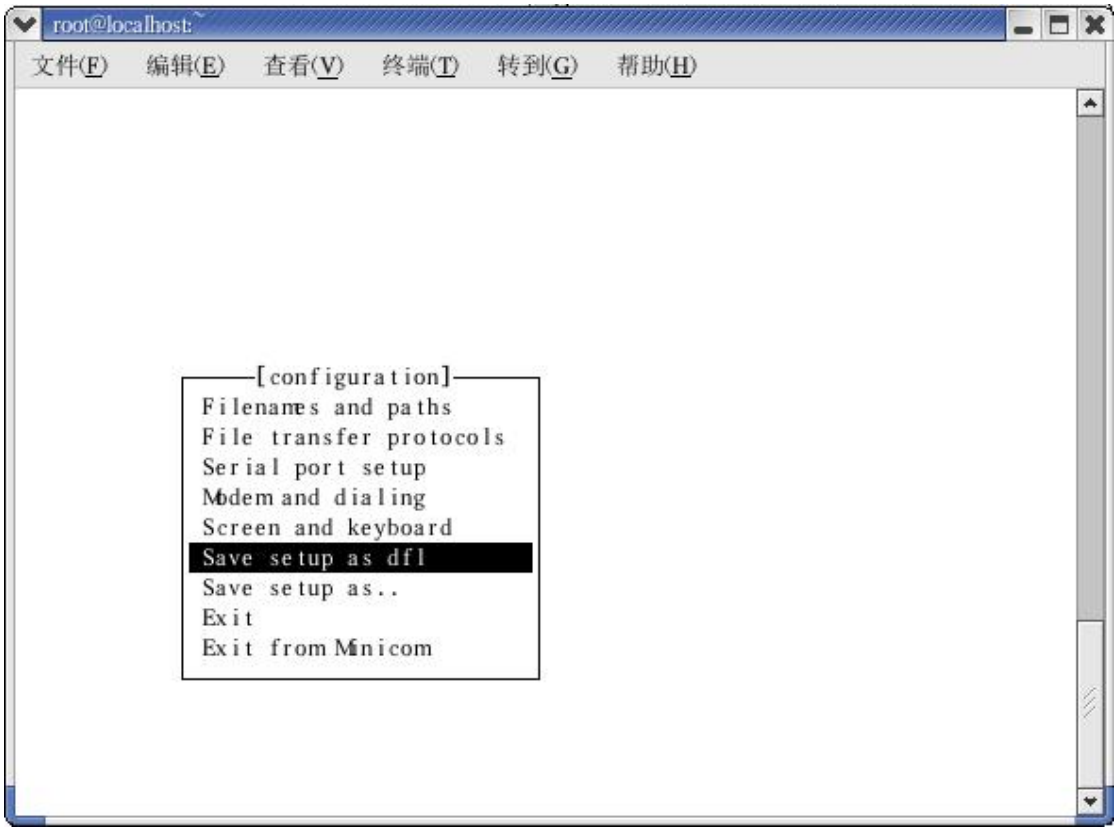
选择 “Serial port setup”项目。

按下 “A” 键设置 “Serial Device”，然后输入 PC 上接入的串行端口号（例如采用 COM1，则写入 /dev/ttyS0，如果是 COM2，写入/dev/ttyS1。）按下“E”键，设置波特率/奇偶位/位数，设置结构为 115200 8N1。按下“F”键，将“Hardware Flow Control”设置为 NO。最终结果如下（采用 COM1 口的例子）：

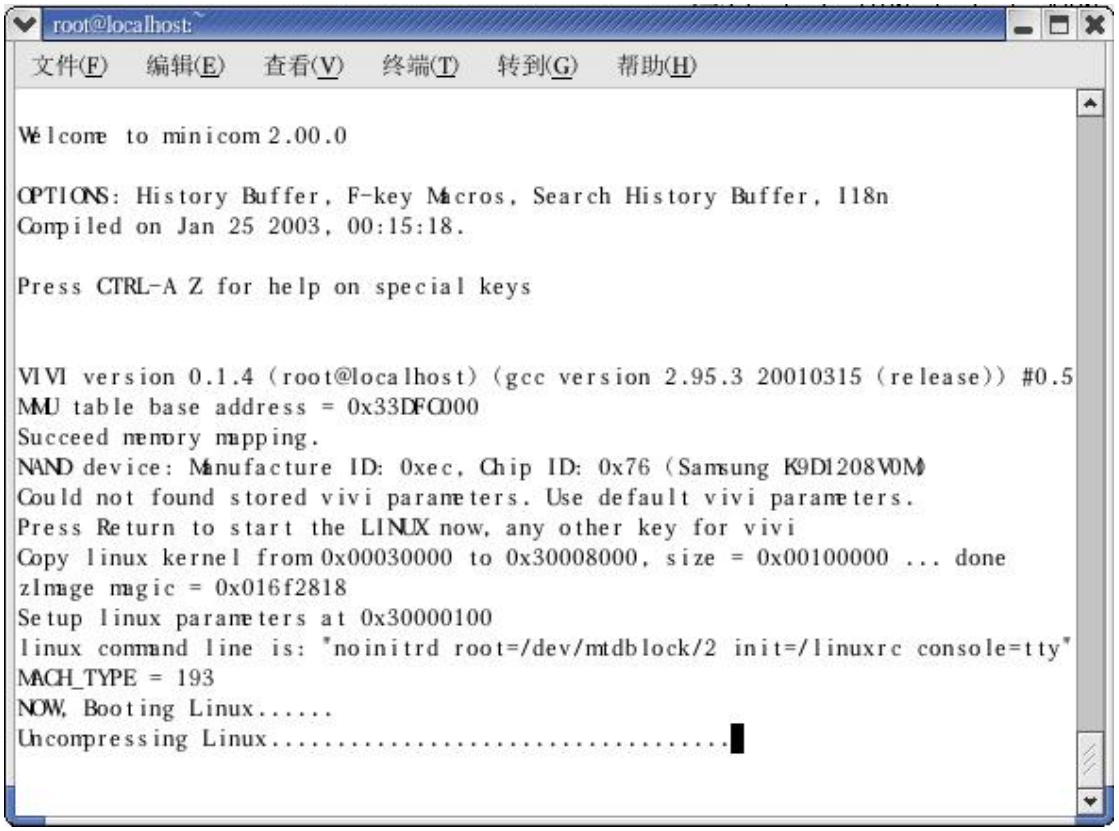


按回车键退出设置。

选择“Save setup as dfl”，并回车保存设置。



这时，复位开发板，如果你的开发板已经预装了 linux，在 minicom 上将看到以下的信息。



2．2 JTAG 的使用

采用 JTAG 来固化 bootloader。JTAG 连接由并口线与 JTAG 小板构成。如果你想重新固化 Bootloader，用 ARMSYS2410 配套的并口线连接 PC 机和 JTAG 小板，再用 20 针排线连接 JTAG 小板和 ARMSYS2410 开发板。采用 JTAG 烧录 bootloader 的详细过程，参考第 4 节。

2．3 网络通讯的使用

网络通讯可以采用以太网交叉线来直接连接 PC 机和开发板，也可以用直通网线来连接 Hub 和开发板。网络通讯在 NFS 方式 mount 主机、root 文件系统更新、以及进行各种网络应用调试时必须用到。关于这些内容，请参考第 6 节、第 7 节和附录 C。

在 ARMSYS2410 中，我们一般假设主机的 IP 地址为 192.168.253.1，开发板的目标地址设定为 192.168.253.8。

3．软件环境

3．1 安装交叉编译器

（1）在编译linux之前，要先安装交叉编译toolchain，安装包位于光盘中的\Linux \ toolchain\ cross-2.95.3.tar.bz2。

在/usr/local 目录下建立名为arm的目录，进入该目录，执行解包：

cd /usr/local/arm
tar xjvf /mnt/cdrom/linux/toolchain/cross-2.95.3.tar.bz2

（2）编辑/etc/profile，找到 pathmunge /usr/local/sbin 在其下面添加一行：

pathmunge /usr/local/arm/2.95.3/bin

这样，内核或其他应用程序均可用 arm-linux- 来指定使用该交叉编译器。

3．2 准备好 Linux 内核源码包

3．2．1 解压

在根目录下建立 armsys2410 目录，进入该目录，执行解包：

mkdir armsys2410
cd armsys2410
tar xzvf /mnt/cdrom/linux/kernel/kernel-armsys-xxxxxxxx.tgz

这里 xxxxxxxx 的内容一般是日期，根据版本的不同而不同。

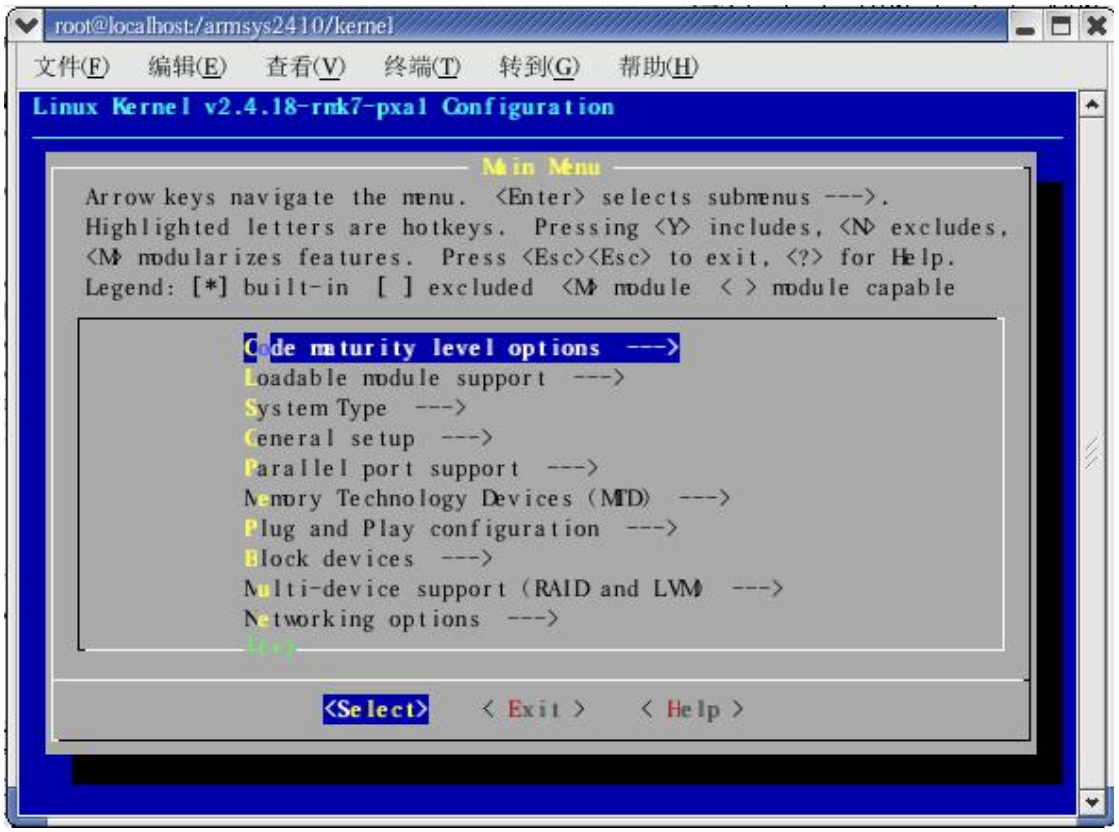
注意：如果您是套餐 3/4/5 的用户，请到“ARMSYS2410 套餐 x 附加内容”目录下得到内核源码包，具体位置请查看该目录下的 readme 文件。一般放在 linuxkernel_*目录下。
此时在 armsys2410 下产生一个 kernel 目录，就是 linux 的内核源码包。

3．2．2 配置

命令行如下：

cd /armsys2410/kernel
make menuconfig

终端中显示：

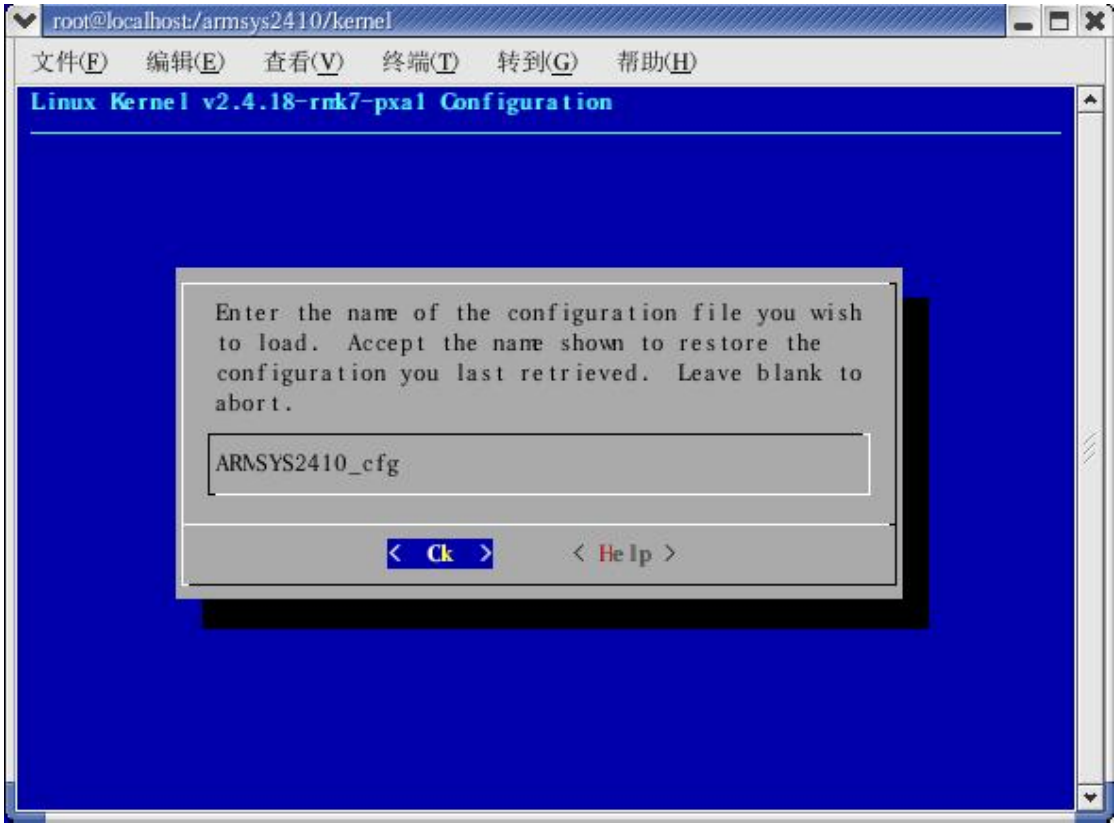

root@localhost:/armsys2410/kernel
文件(F) 编辑(E) 查看(V) 终端(T) 转到(G) 帮助(H)
Linux Kernel v2.4.18-rmk7-pxa1 Configuration
Main Menu
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help.
Legend: [*] built-in [ ] excluded <M> module < > module capable
Code maturity level options --->
Loadable module support --->
System Type --->
General setup --->
Parallel port support --->
Memory Technology Devices (MTD) --->
Plug and Play configuration --->
Block devices --->
Multi-device support (RAID and LVM) --->
Networking options --->
<Select> < Exit > < Help >

（1）选中 Load an Alternate Configuration File，敲空格进入：

root@localhost:/armsys2410/kernel
文件(F) 编辑(E) 查看(V) 终端(T) 转到(G) 帮助(H)
Linux Kernel v2.4.18-rmk7-pxa1 Configuration
Main Menu
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help.
Legend: [*] built-in [ ] excluded <M> module < > module capable
File systems --->
Console drivers --->
Sound --->
Multimedia Capabilities Port drivers --->
USB support --->
Bluetooth support --->
Kernel hacking --->

Load an Alternate Configuration File
Save Configuration to an Alternate File
<Select> < Exit > < Help >

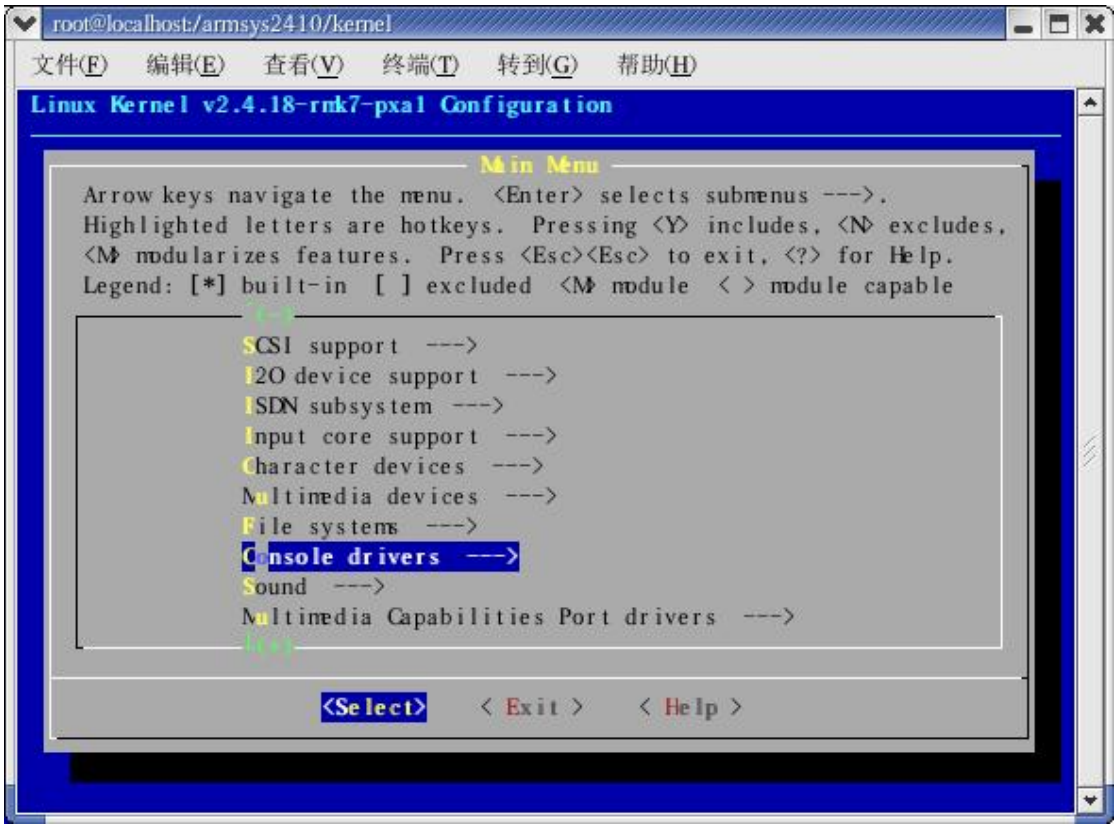
输入 ARMSYS2410_cfg，点击 OK：



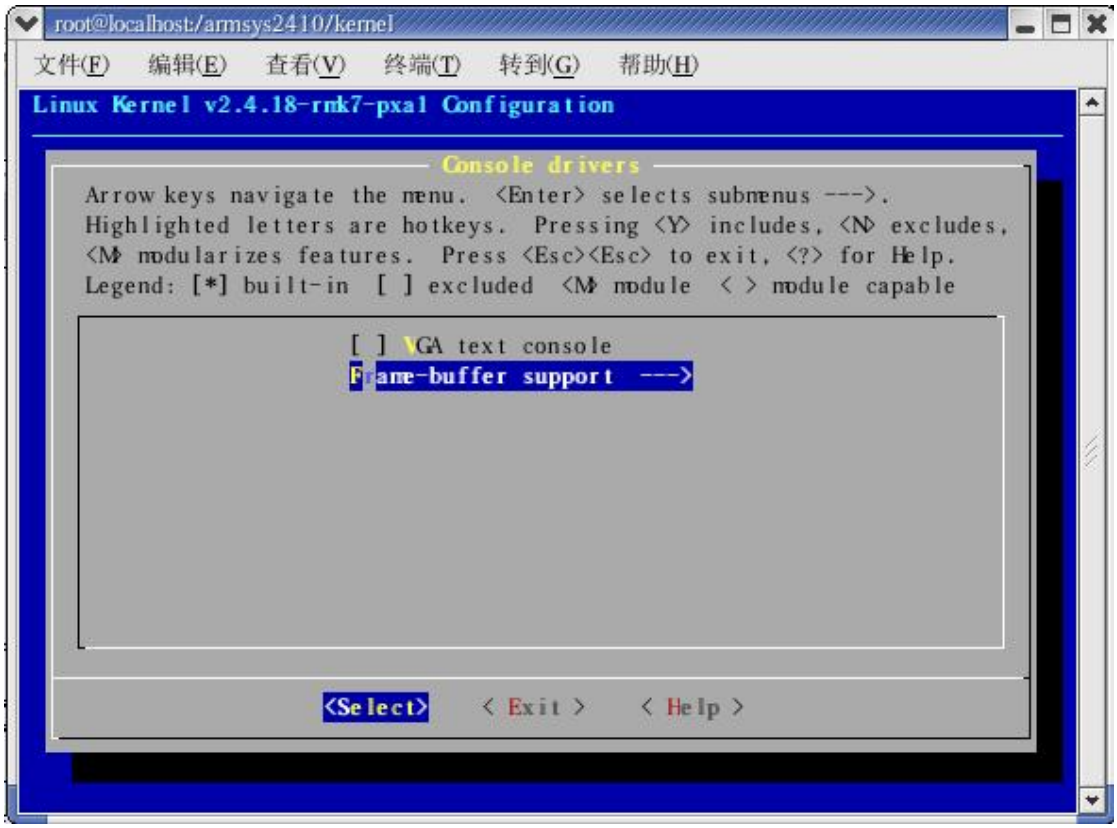
这样就完成了针对 ARMSYS2410 的基本配置。

如果您购买的是带有液晶屏和触摸屏的 ARMSYS2410 套餐，则还要配置一下您的 LCD 和触摸屏规格的类型。如果您购买的是单板机，则跳过（2）和（3）。

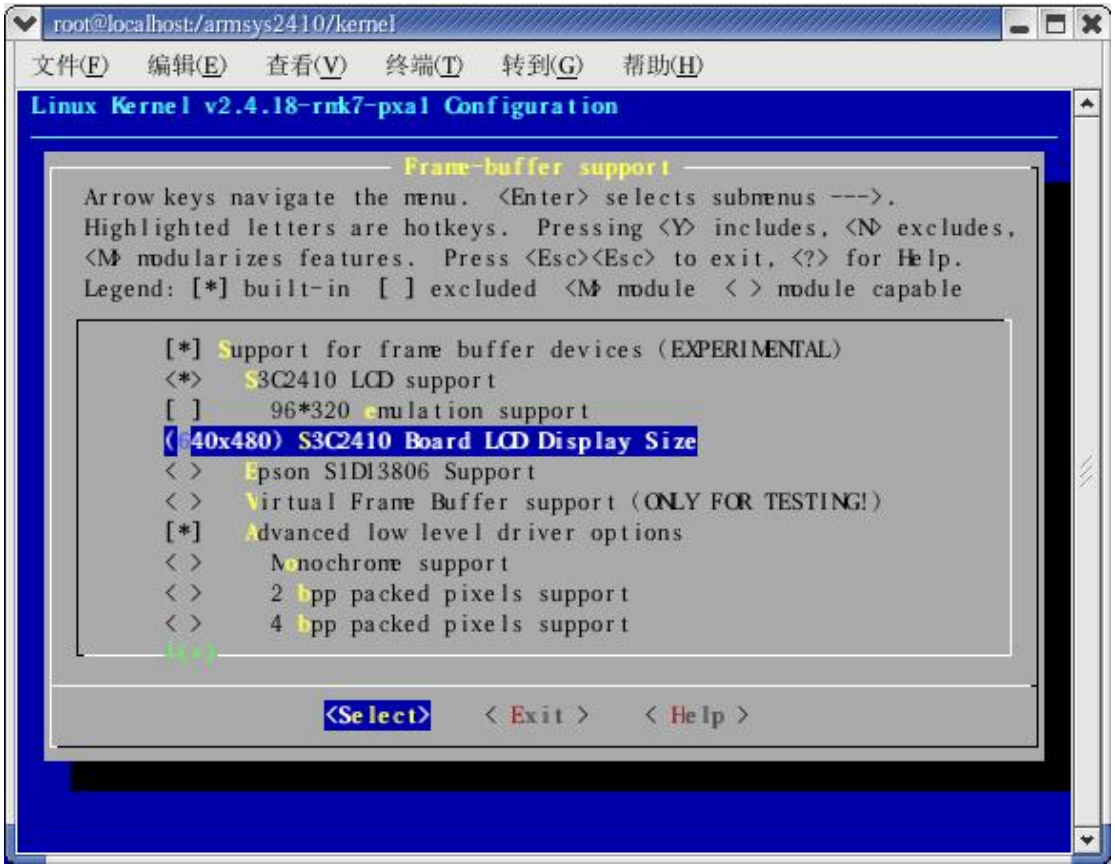
（2）LCD 相关选项在 Console drivers →下，敲空格进入：



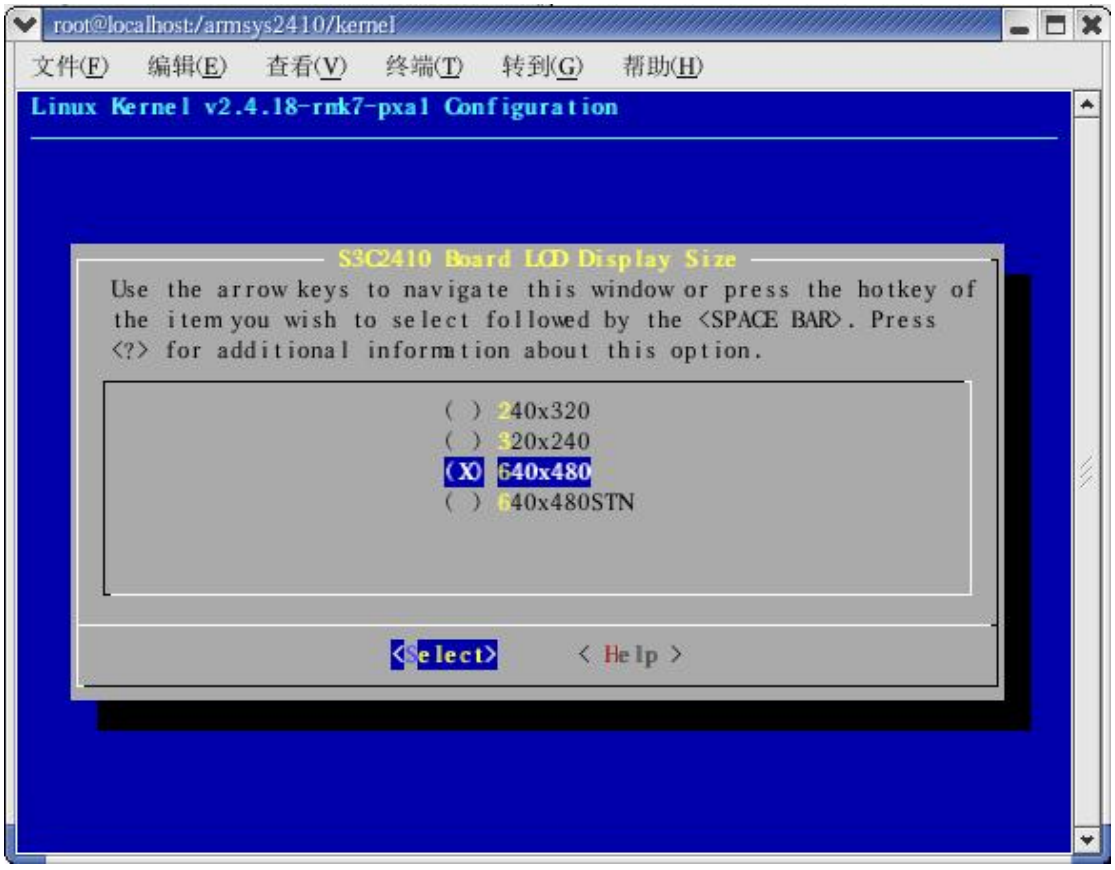
进入 Frame-buffer support --->


root@localhost:/armsys2410/kernel
文件(F) 编辑(E) 查看(V) 终端(T) 转到(G) 帮助(H)
Linux Kernel v2.4.18-rmk7-pxa1 Configuration
Console drivers
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help.
Legend: [*] built-in [ ] excluded <M> module < > module capable
[ ] VGA text console
Frame-buffer support --->
<Select> < Exit > < Help >

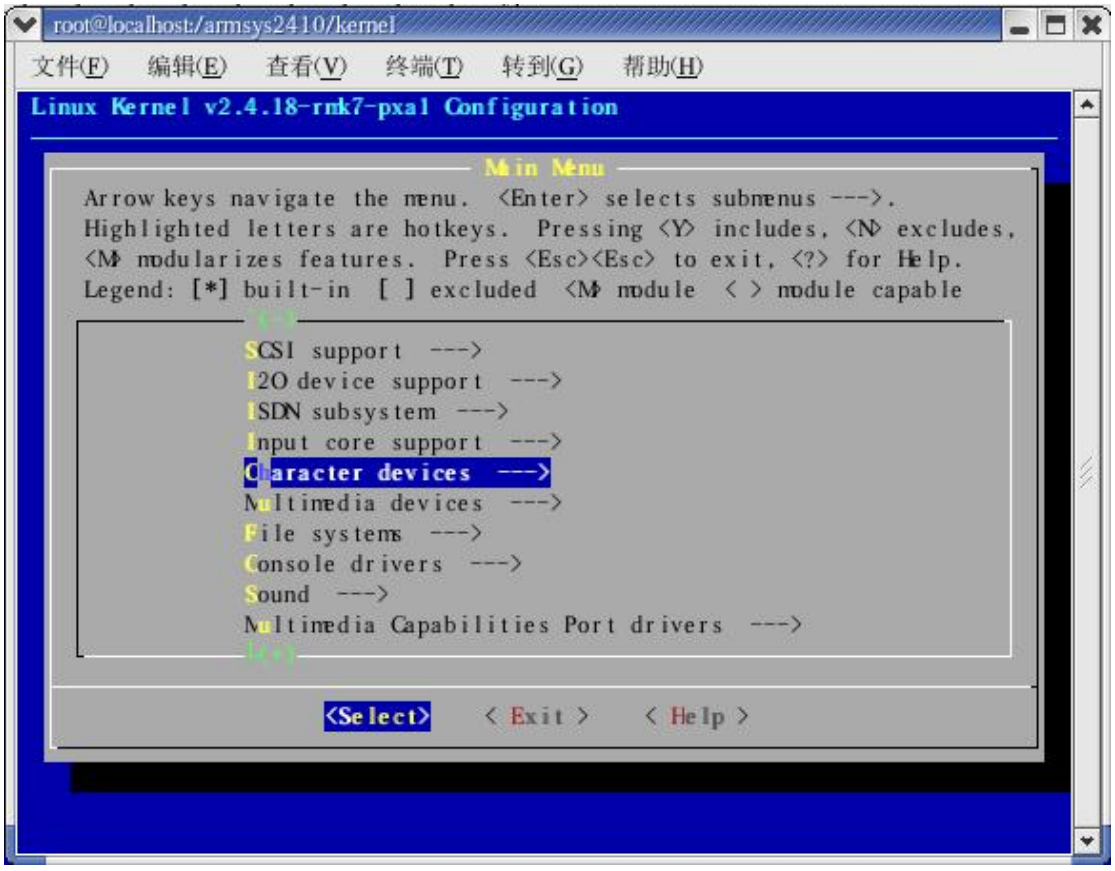
选择 S3C2410 Board LCD Display Size：


root@localhost:/armsys2410/kernel
文件(F) 编辑(E) 查看(V) 终端(T) 转到(G) 帮助(H)
Linux Kernel v2.4.18-rmk7-pxa1 Configuration
Frame-buffer support
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help.
Legend: [*] built-in [ ] excluded <M> module < > module capable
[*] Support for frame buffer devices (EXPERIMENTAL)
<*> S3C2410 LCD support
[ ] 96*320 emulation support
(240x480) S3C2410 Board LCD Display Size
< > Epson S1D13806 Support
< > Virtual Frame Buffer support (ONLY FOR TESTING!)
[*] Advanced low level driver options
< > Monochrome support
< > 2 bpp packed pixels support
< > 4 bpp packed pixels support
<Select> < Exit > < Help >

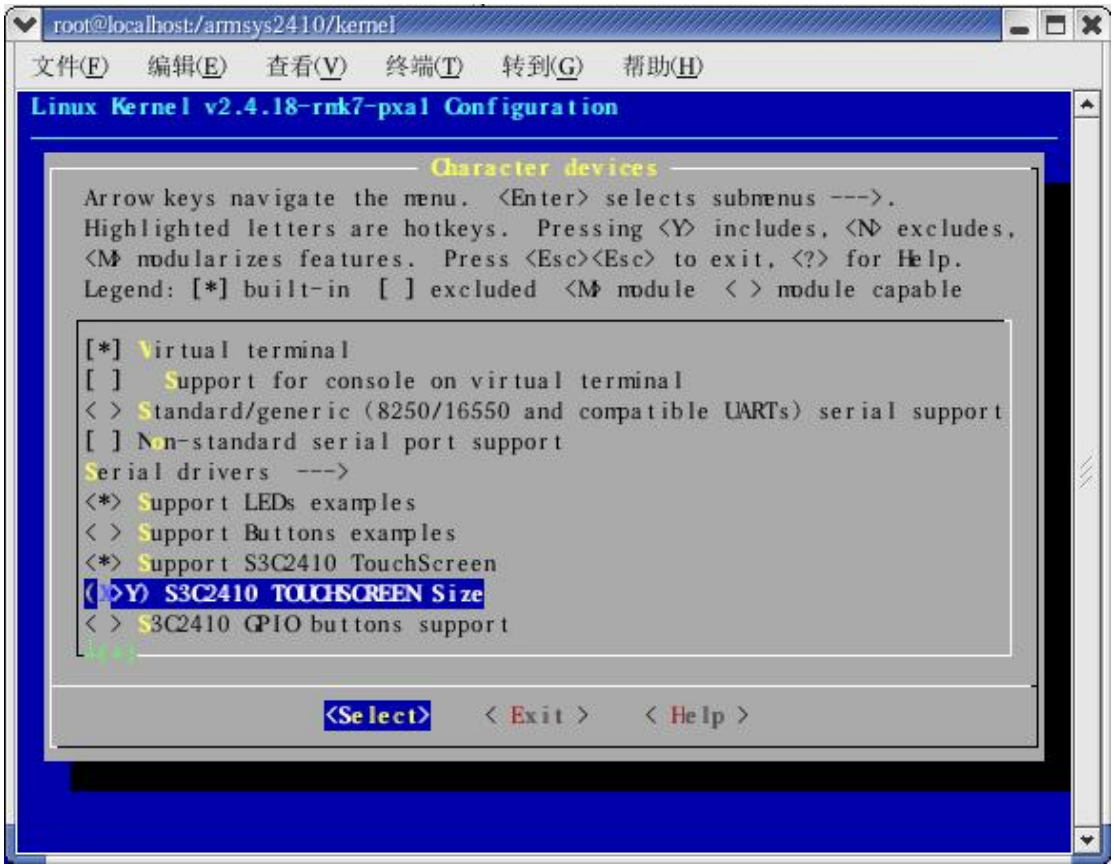
根据你的 LCD 类型和解析率选择特定的项目：


root@localhost:/armsys2410/kernel
文件(F) 编辑(E) 查看(V) 终端(T) 转到(G) 帮助(H)
Linux Kernel v2.4.18-rmk7-pxa1 Configuration
S3C2410 Board LCD Display Size
Use the arrow keys to navigate this window or press the hotkey of the item you wish to select followed by the <SPACE BAR>. Press <?> for additional information about this option.
( ) 240x320
( ) 320x240
(X) 640x480
( ) 640x480STN
<Select> < Help >

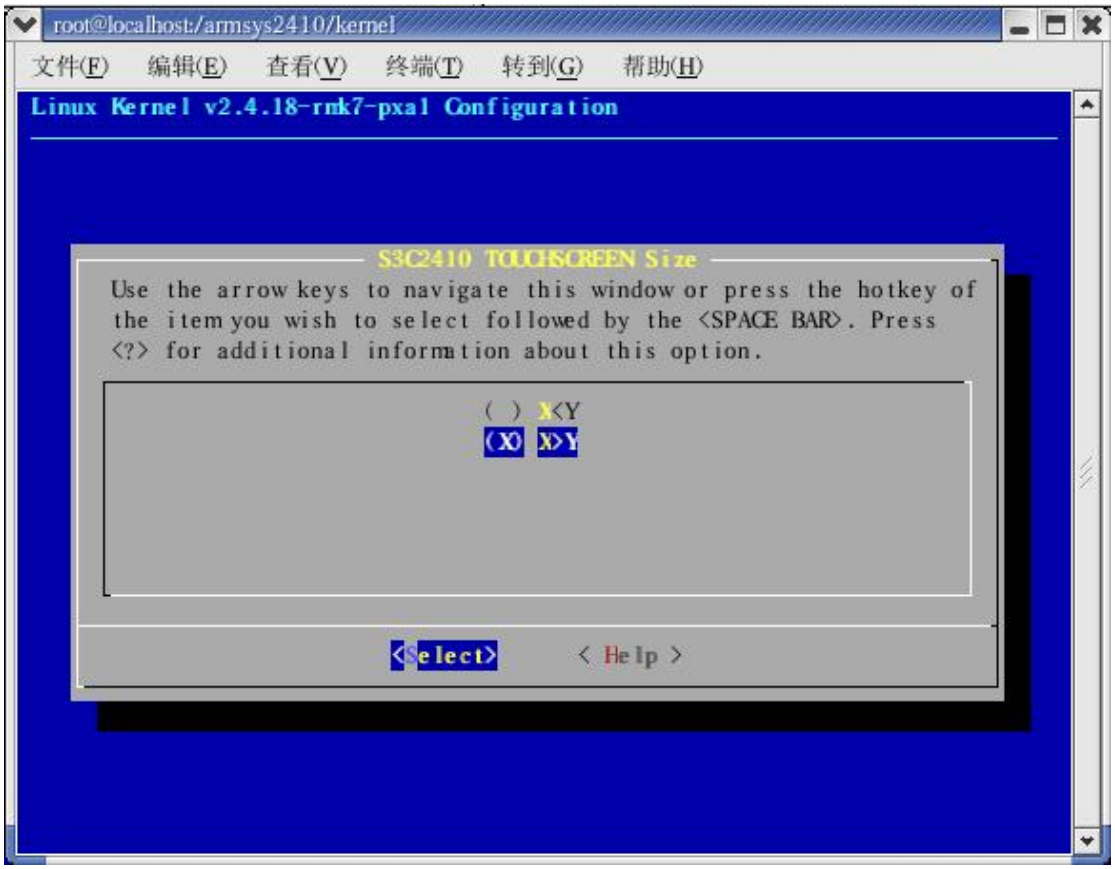
（3）触摸屏选项在 Character devices→下，退出到 Main Menu 点击进入：


root@localhost:/armsys2410/kernel
文件(F) 编辑(E) 查看(V) 终端(T) 转到(G) 帮助(H)
Linux Kernel v2.4.18-rmk7-pxa1 Configuration
Main Menu
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help.
Legend: [*] built-in [ ] excluded <M> module < > module capable
SCSI support --->
I2O device support --->
ISDN subsystem --->
Input core support --->
Character devices --->
Multimedia devices --->
File systems --->
Console drivers --->
Sound --->
Multimedia Capabilities Port drivers --->
<Select> < Exit > < Help >

选择 S3C2410 TOUCHSCREEN Size：



根据你的触摸屏规格（横放/竖放）选择特定的项目：



X>Y 代表触摸屏横放；Y>X 代表触摸屏竖放。

设定好后，可以通过 Save Configuration to an Alternate file 保存你刚刚的配置。

3．2．3 编译

命令行如下：

make clean
make dep
make zImage

编译结束后将在 kernel/arch/arm/boot 目录下得到 linux 内核压缩映像文件：zImage。

3．3 准备好 root 文件系统

光盘资料中提供了两套文件系统，

- root_nfs 目录：NFS 文件系统，保存在作为主机的 PC 上，主要用于 NFS 调试和系统更新；
- root_yaffs 目录：固化在开发板的 Nandflash 中的 YAFFS 文件系统内容。

将光盘中 Linux/rootfilesystem/root_nfs/下的压缩文件解压，得到 root_nfs 目录。然后将该目录更名为 root，拷贝到/armsys2410/下，这样就准备好了共享目录。

注意在这个/armsys2410/root 中有一个的压缩包 root_armsys.tgz，这就是要固化到 Nandlflash 中去的 yaffs 文件系统。如果您是购买套餐的用户，请到“ARMSYS2410 套餐 x 附加内容”目录下得到文件系统压缩包（**具体可以看该目录下的 readme 文件，一般保存在 linuxrootfs*目录下**），拷贝到/armsys2410/root/中去，改名为 root_armsys.tgz。

在后面的章节中，我们将使用到这两个文件系统。

3．4 准备好 vivi Bootloader

Bootloader 在嵌入式系统中扮演着重要的角色。它主要完成以下任务：

- 将内核从 flash 存储器拷贝到 RAM，并执行内核；
- 初始化硬件；
- 提供写入 flash 存储器（固化程序）的功能；
- 提供向目标板发送命令和通知用户目标板状态的界面；

目前我们提供两种 bootloader：u-boot 和 vivi。

为 u-boot 我们专门提供了文档《如何使用 ARMSYS2410 配套的 u-boot》，请到 linux/u-boot/目录下查看。本文不再描述。

Vivi Bootloader 是由 Mizi 公司开发的一个 Bootloader，目前仅支持通过串口进行下载，因此在使用前必须用串口线连接 PC 机的串口和开发板的 COM1 口，并在主机 linux 下打开

minicom 使用。

将光盘中的 vivi 目录拷贝到 armsys2410/下：

cp –av /mnt/cdrom/linux/vivibootloader/vivi /armsys2410/vivi

如果你需要对 vivi 进行修改，则也同样需要重新编译它。

配置 vivi：

cd /armsys2410/vivi

make menuconfig

同样选中 Load an Alternate Configuration File，载入默认配置文件：

Enter the name of the configuration file you wish to load. Accept the name shown to restore the configuration you last retrieved. Leave blank to abort.
arch/def-configs/smdk2410
< Ok >
< Help >

选择 OK，敲“ESC”退出，出现以下界面：

选择 Yes 退出。

然后：

make

即可。如果编译成功，在 vivi 目录下将产生名为 vivi 的二进制文件。

在后面的章节中，我们将会将 vivi Bootloader、kenerl 映像和 root 文件系统下载到开发板中去。因此，为了方便我们将用到的映像文件都存放在一个目录下，例如放到/armsys2410/images/。在我们的光盘中也有一个包含默认映像文件的目录，如果要使用它来恢复系统，可以将它拷贝到硬盘中：

cd armsys2410

cp –av /mnt/cdrom/linux/images images

4. Vivi Bootloader 固化到 ARMSYS2410

一般预装 linux 的开发板都已经固化好了 vivi bootloader。如果你不想更新 bootloader，则不需要阅读这一节。

用 ARMSYS2410 套件中提供的并口线连接 JTAG 小板，再用 20 针排线连接 JTAG 小板和 ARMSYS2410 开发板。令开发板上电。将烧录工具软件（Jflash_Linux 目录下）Jflash-s3c2410 和 vivi 两个文件拷贝/armsys2410/Jflash 下。

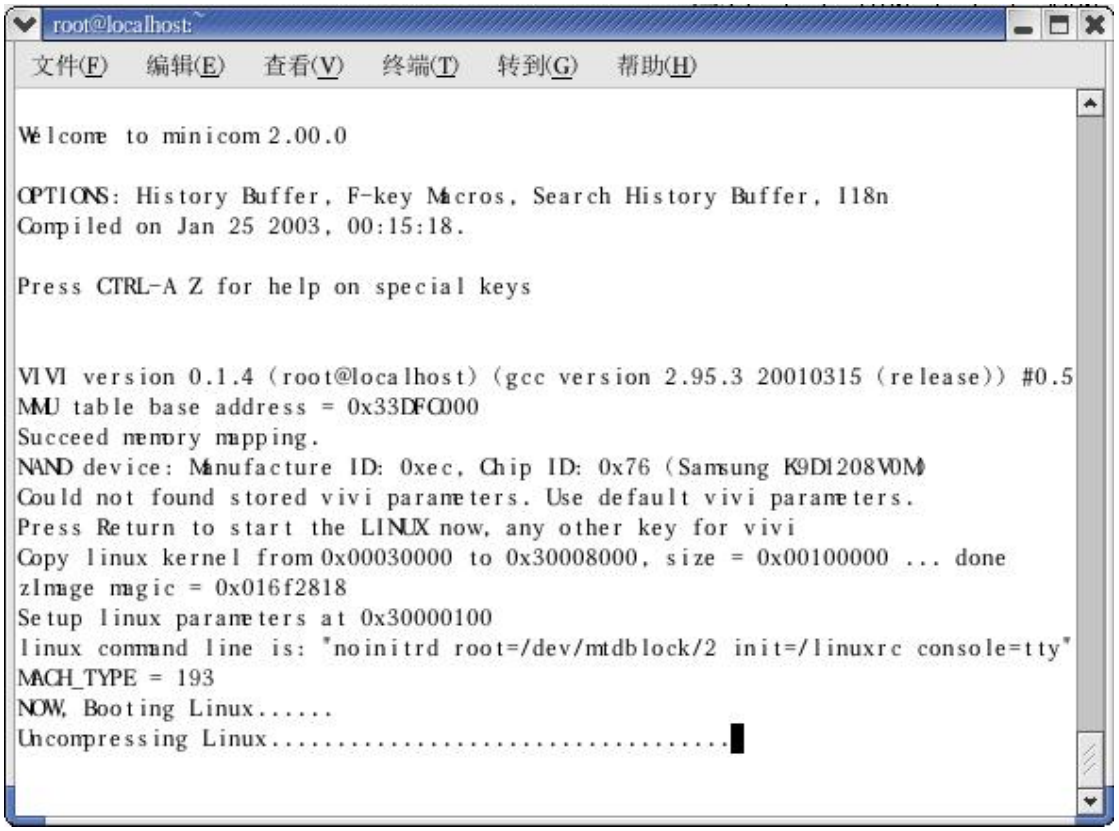
如果更新了 vivi 则将你自己的 vivi 拷贝到这个目录下。开始烧录：

cd Jflash
./Jflash-s3c2410 vivi /t=5

终端上出现：

```
+-----------------------------------+
| SEC JTAG FLASH(SJF) v 0.11    +
| modified by MIZI 2002.7.13    +
+-----------------------------------+
> flashType=5
> S3C2410X(ID=0x0032409d) is detected.
> K9S1208 is detected. ID=0xec76

K9S1208 NAND Flash JTAG Programmer Ver 0.0
0:K9S1208 Program 1:K9S1208 Pr BlkPage 2: Exit

Select the function to test :0 :这里输入“0”.

[SMC(K9S1208) NAND Flash Writing Program]

Source size: 0xe4af

Available target block number: 0~4095
Input target block number:0 : 这里输入“0”.
target start block number =0
target size (0x4000*n) =0x10000
STATUS:Epppppppppppppppppppppppppppppppp
Epppppppppppppppppppppppppppppppp
Epppppppppppppppppppppppppppppppp
Epppppppppppppppppppppppppppppppp

K9S1208 NAND Flash JTAG Programmer Ver 0.0
0:K9S1208 Program 1:K9S1208 Pr BlkPage 2: Exit

Select the function to test :2 : 输入“2”退出烧录.
```

这样，vivi 就被成功地烧录到了 flash 中。关闭开发板电源，拔除 JTAG 小板，然后重新复位开发板，可以从 minicom 上看到如下输出信息：



如果 flash 中还没有固化内核，则系统运行到“Uncompress Linux……”即告终止了。下面我们来看如何固化内核。

5. Kernel（Linux 内核）固化到 ARMSYS2410

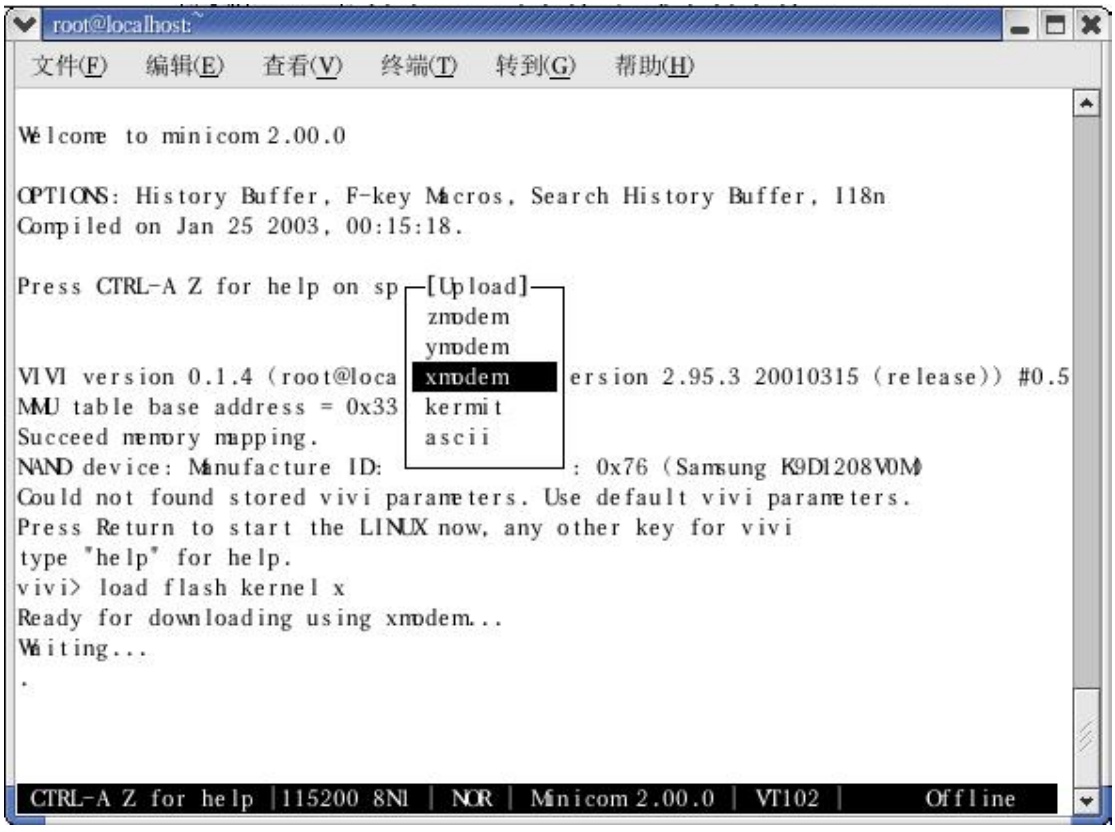
一般预装 linux 的开发板都已经固化好了 Linux kernel。如果您不想更新 kernel，则不需要阅读这一节。

（1）在 PC 机键盘上按住空格键，然后按下 ARMSYS2410 开发板上的复位键，vivi 在 minicom 上的输出如上图所示。

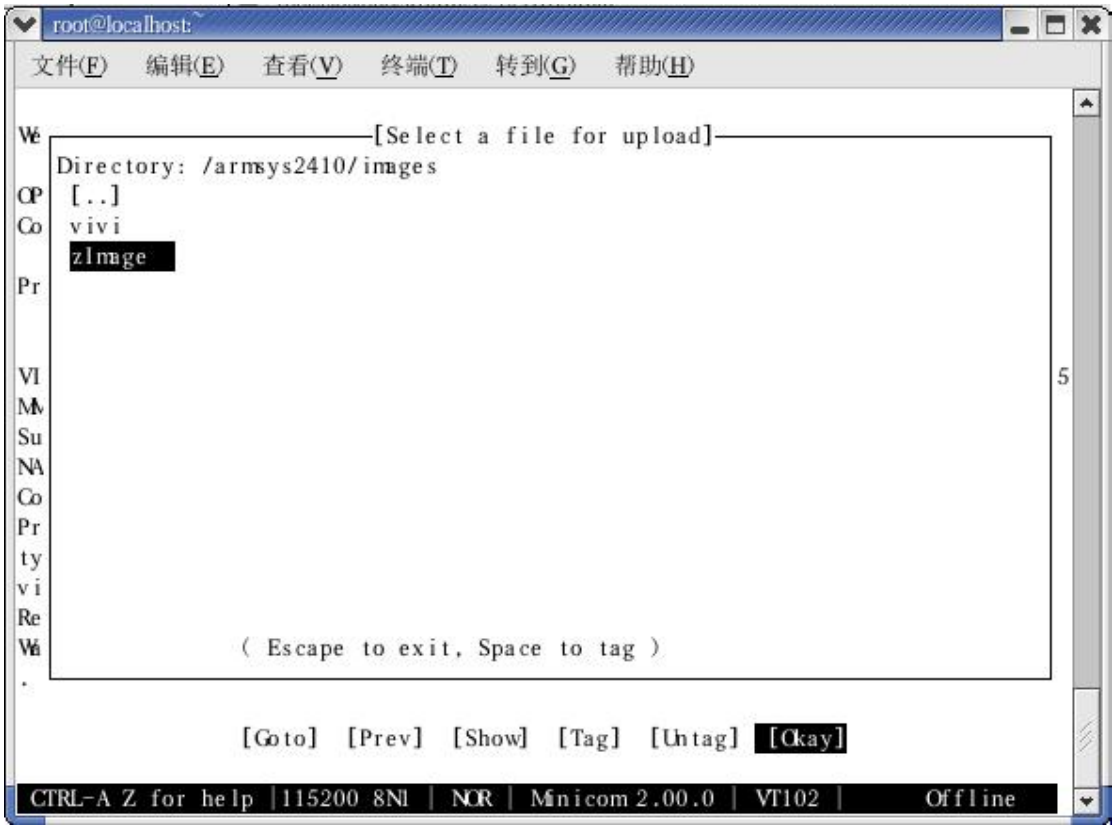
（2）在 vivi>后输入命令行：

load flash kernel x

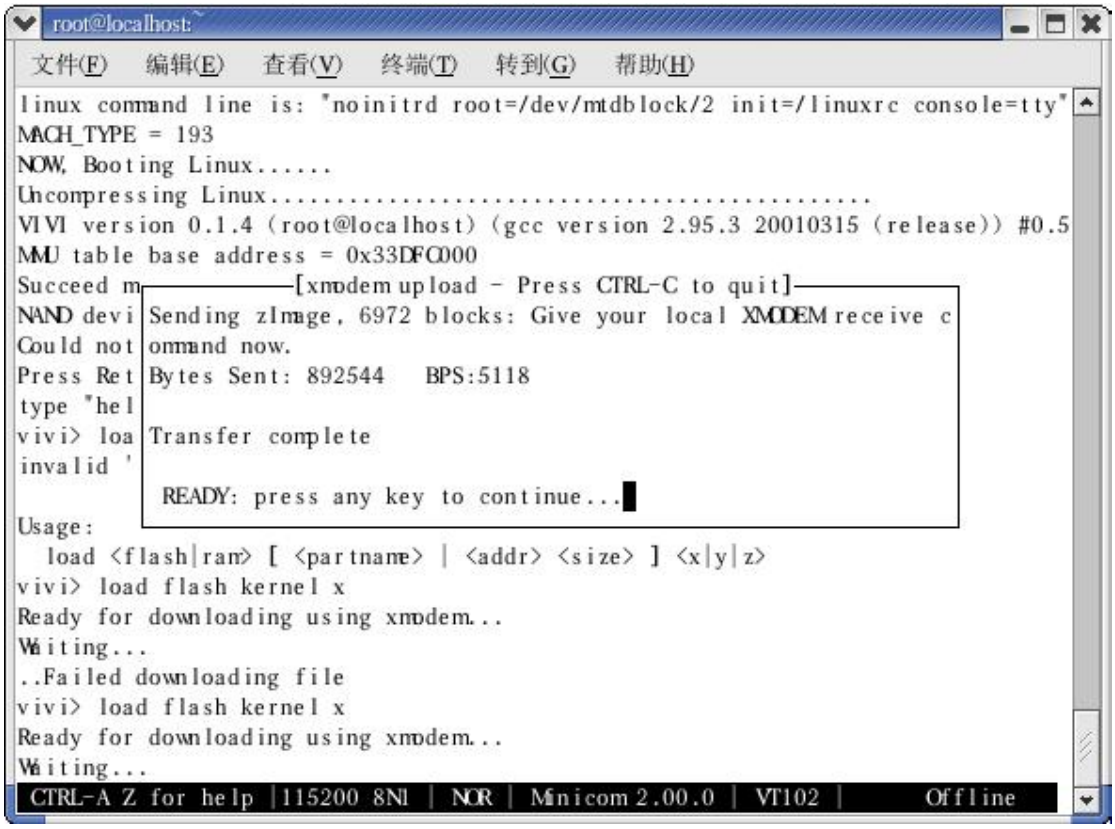
（3）然后按下 Ctrl 键和 A 键（同时按住一会儿），然后松开，再按一下 S 键。这时弹出对话框：



（4）进入/armsys2410/images 目录（或其他你存放 kernel 映像文件的目录，例如 kernel/arch/arm/boot/），选中 zImage：


root@localhost:~
文件(F) 编辑(E) 查看(V) 终端(T) 转到(G) 帮助(H)
[Select a file for upload]
Directory: /armsys2410/images
[..]
vivi
zImage
(Escape to exit, Space to tag)
[Goto] [Prev] [Show] [Tag] [Untag] [Okay]
CTRL-A Z for help |115200 8N1 | NOR | Minicom 2.00.0 | VT102 | Offline

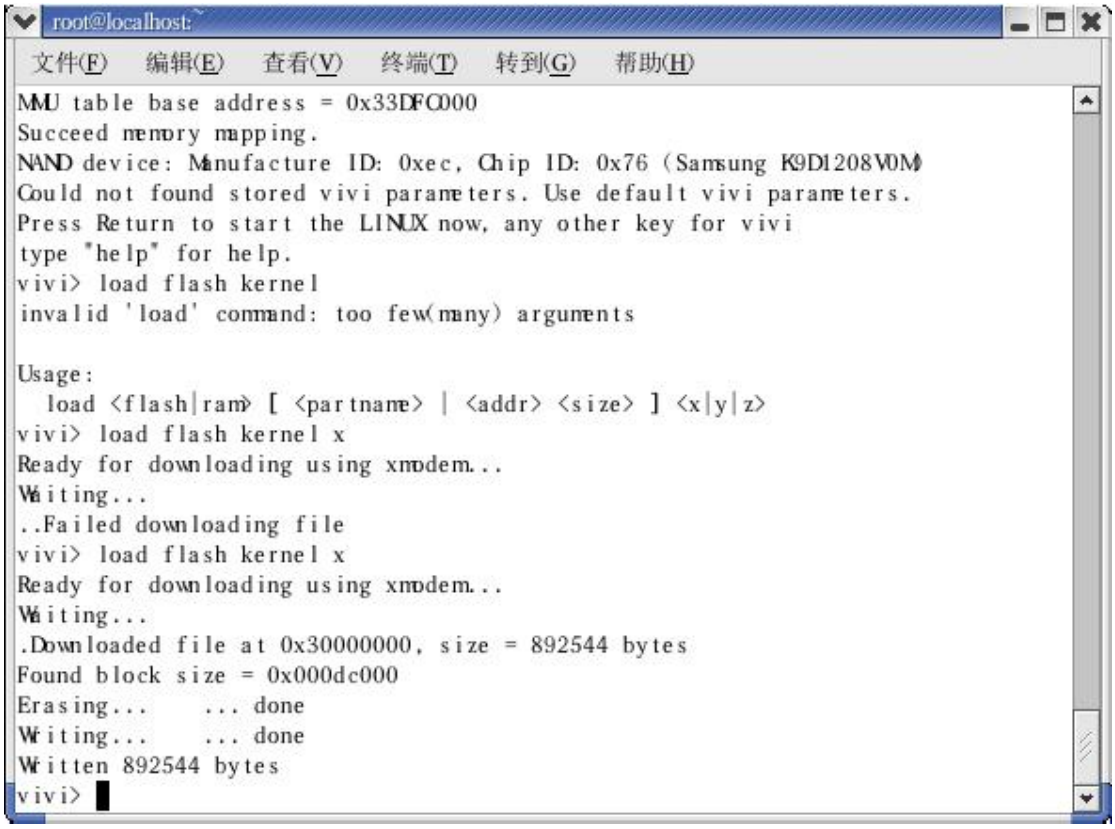
（5）按空格，开始传输：


root@localhost:~
文件(F) 编辑(E) 查看(V) 终端(T) 转到(G) 帮助(H)
linux command line is: "noinitrd root=/dev/mtdblock/2 init=/linuxrc console=tty"
MACH_TYPE = 193
NOW, Booting Linux.....
Uncompressing Linux.....
VIVI version 0.1.4 (root@localhost) (gcc version 2.95.3 20010315 (release)) #0.5
MMU table base address = 0x33DFC000
[xmodem upload - Press CTRL-C to quit]
Sending zImage, 6972 blocks: Give your local XMODEM receive command now.
Bytes Sent: 892544 BPS:5118
Transfer complete
READY: press any key to continue...
Usage:
load <flash|ram> [<partname> | <addr> <size>] <x|y|z>
vivi> load flash kernel x
Ready for downloading using xmodem...
Waiting...
..Failed downloading file
vivi> load flash kernel x
Ready for downloading using xmodem...
Waiting...
CTRL-A Z for help |115200 8N1 | NOR | Minicom 2.00.0 | VT102 | Offline

（6）如果长时间没有开始传输，按下 Ctrl+C 键，然后在 vivi 下也按下 Ctrl＋C 键取消这次

传输。重新执行第（2）步～第（5）步。

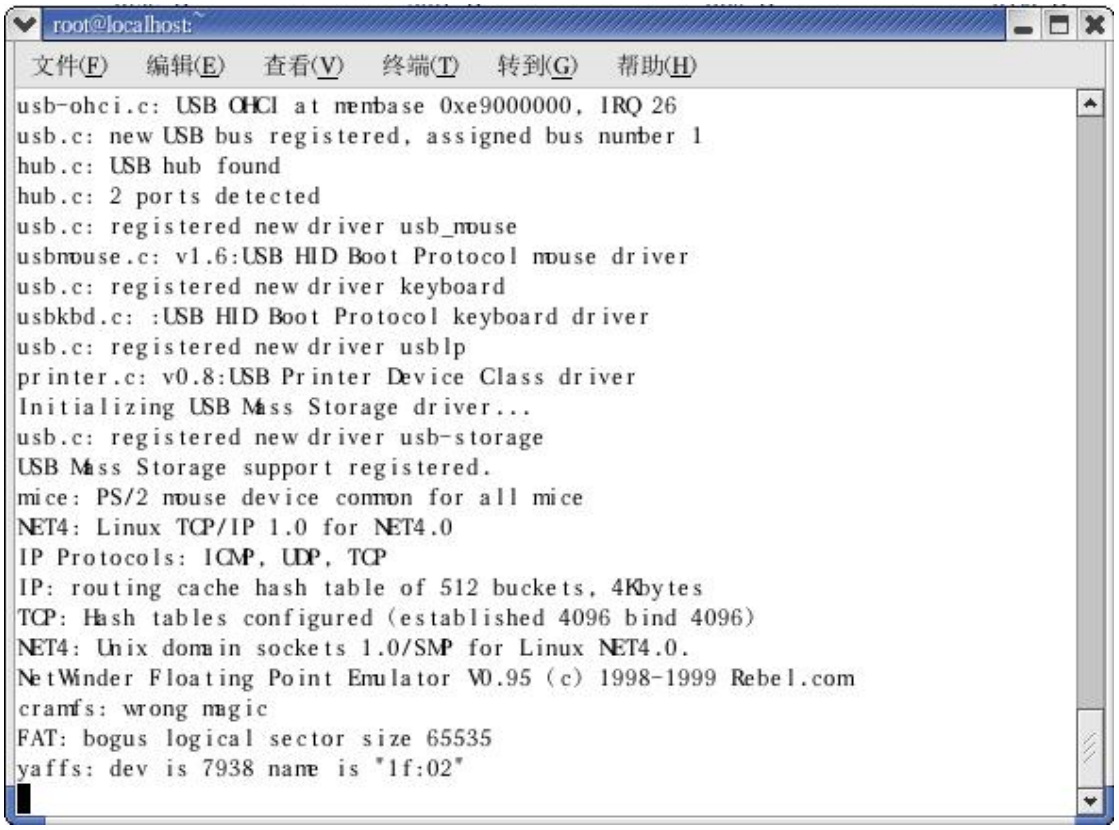
（7）传输完成后，有一个写入的过程。等待一会儿，写入过程结束：



（8）为了验证内核的写入正确，vivi>下输入：

boot

如果一切顺利，minicom 上输出启动内核的信息：



6．NFS 方式 mount 主机

6．1 在主机上启动 nfs 服务

启动 nfs 服务之前，必须准备好共享目录，
在 3.3 节中已经提到两个文件系统，这里再次说明一下。

将光盘中 Linux/rootfilesystem/root_nfs/下的压缩文件解压，得到 root_nfs 目录。然后将该目录更名为 root，拷贝到/armsys2410/下，这样就准备好了共享目录。

注意在这个/armsys2410/root 中有一个的压缩包 root_armsys.tgz，这就是要固化到 Nandlflash 中去的 yaffs 文件系统。如果您是购买套餐的用户，请到“ARMSYS2410 套餐 x 附加内容”目录下得到文件系统压缩包（具体可以看该目录下的 readme 文件，一般保存在 linuxrootfs*目录下），拷贝到/armsys2410/root/中去，改名为 root_armsys.tgz。

（1） 修改/etc/exports 文件的内容，添加一行：

/armsys2410/root *(rw,sync,no_root_squash)

注意：root和*之间有空格！

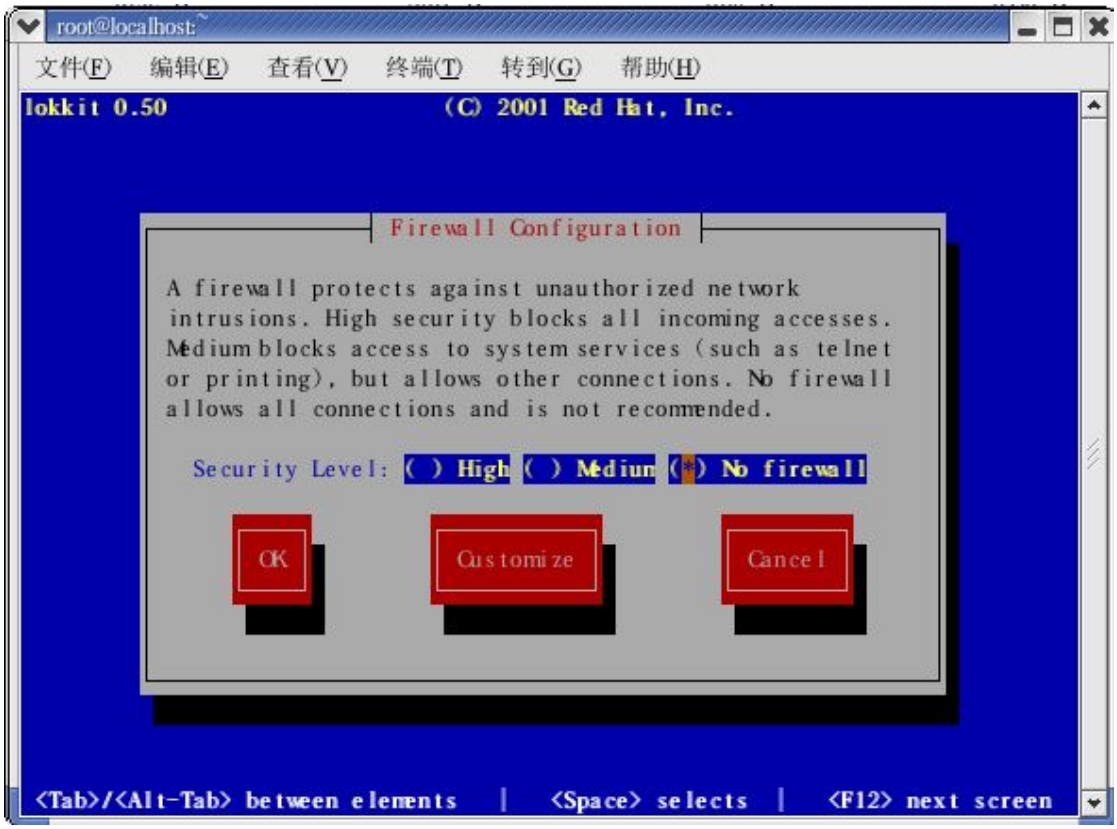
这里/armsys2410/root 代表要共享输出的 root 文件系统的共享目录；*代表所有的客户机都可以挂接此文件系统；rw 代表客户机可以读写许可来挂接它们的根文件系统；no_root_squash 选项允许客户机以主机上的 root 身份挂接根文件系统。

（2） 关闭防火墙

终端下输入：

lokkit

进入防火墙设置，将防火墙设置为无：


root@localhost:~
文件(F) 编辑(E) 查看(V) 终端(T) 转到(G) 帮助(H)
lokkit 0.50 (C) 2001 Red Hat, Inc.
Firewall Configuration
A firewall protects against unauthorized network intrusions. High security blocks all incoming accesses. Medium blocks access to system services (such as telnet or printing), but allows other connections. No firewall allows all connections and is not recommended.
Security Level: ( ) High ( ) Medium (*) No firewall
OK
Customize
Cancel
<Tab>/<Alt-Tab> between elements | <Space> selects | <F12> next screen

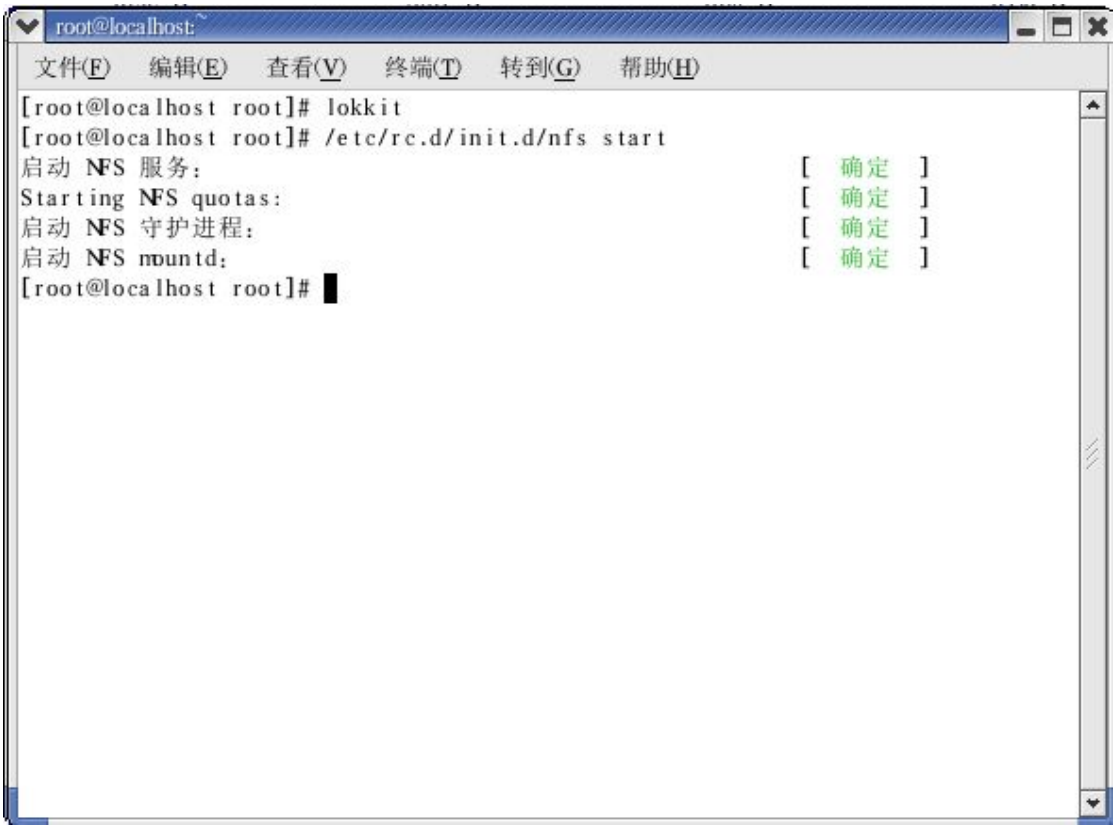
（3）启动 NFS 服务：

输入命令行：

/etc/rc.d/init.d/nfs start

如果之前已经启动了 nfs，则重新启动一次：

/etc/rc.d/init.d/nfs restart



6．2 在目标板上进行设定

（1）设定 Linux 的启动参数
按住 PC 机的空格键，然后复位开发板，此时从 minicom 的输出看到，进入了 vivi 命令行方式。重新设置 Linux 的启动参数如下：

param set linux_cmd_line "console=ttyS0 root=/dev/nfs nfsroot=192.168.253.1:/armsys2410/root ip=192.168.253.8:192.168.253.1:192.168.253.1:255.255.255.0:armsys.hzlitai.com:eth0:off"

命令行必须在一行中输入完。因此我们提供了一个 Linux\rootfilesystem\param_vivi 文本文件，其中已经写好了这一行，只要将它拷贝并粘贴到 vivi>后即可。
注意，以上命令行假定主机地址为 192.168.253.1，ARMSYS2410 开发板的如果你的主机 IP 与此不同，请用 ifconfig 重新设定，或者修改该命令行的内容。
（2）保存参数
在 vivi>下输入：

param save

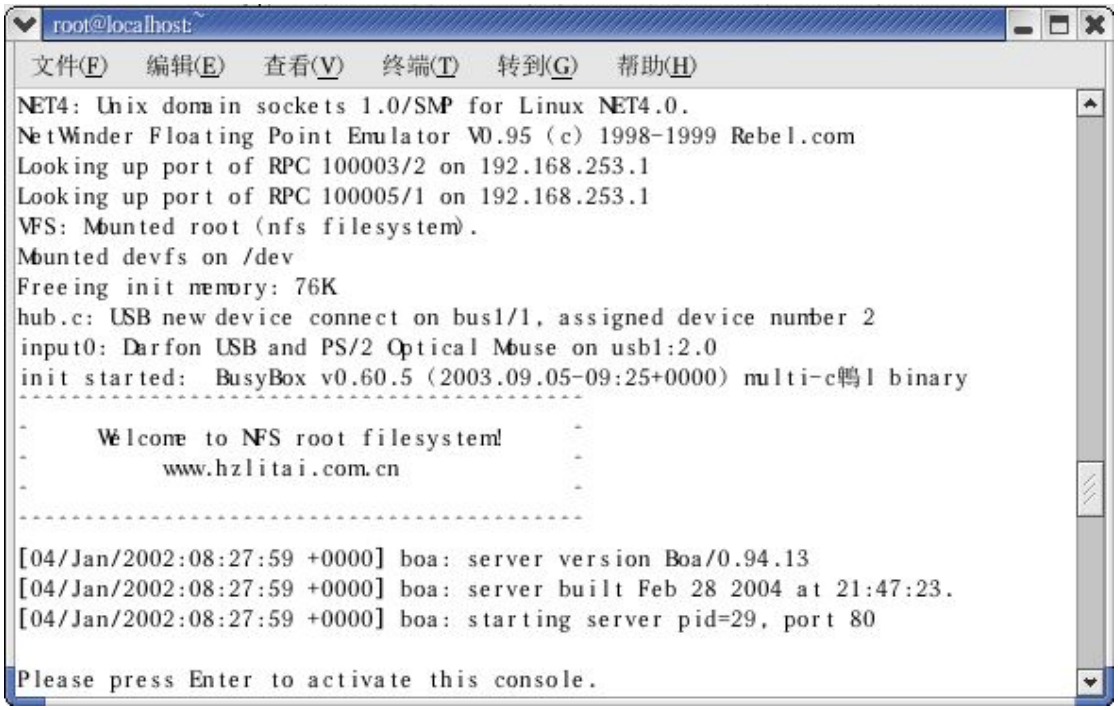
即可保存刚刚设定的参数，这样下一次就不需要重复输入上述命令行了。
Minicom 上的显示如下图所示：

6．3 启动 Linux

确保你已经连接好网线，然后在 vivi>下输入：

boot

输出完整的 linux 启动信息如下：

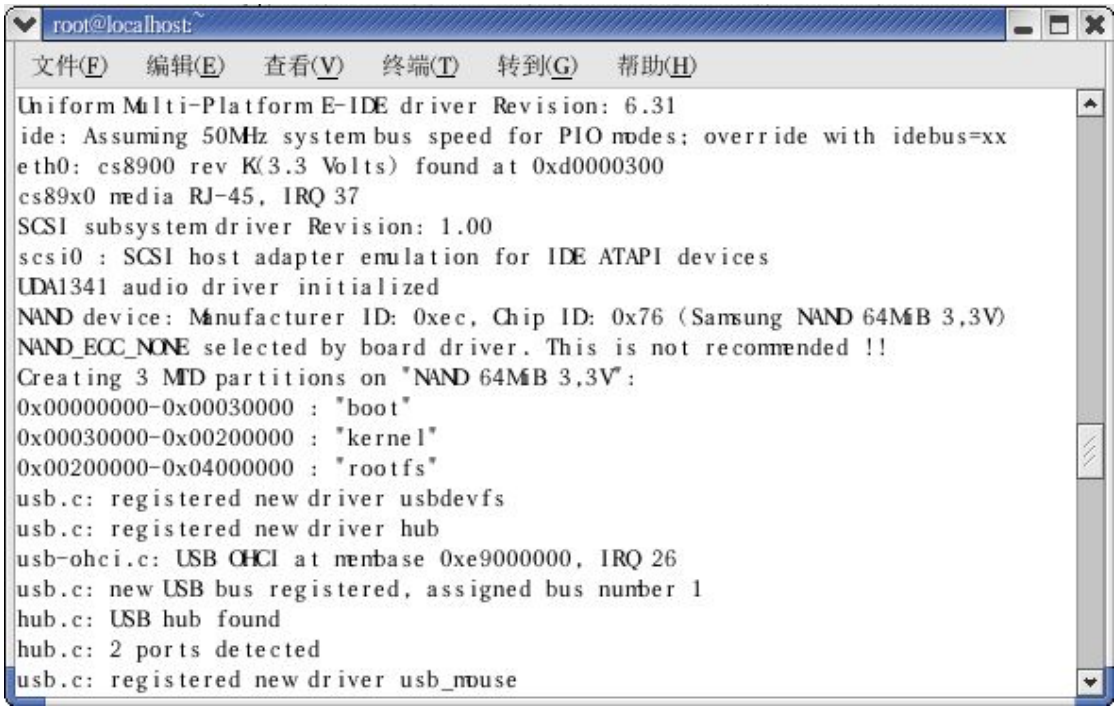


最后部分的信息，说明已经成功地 mount 上了主机的 root 文件系统。

7．NFS 方式更新文件系统

7．1 更新文件系统

在 linux 启动信息中可以注意到：


root@localhost:~
文件(F) 编辑(E) 查看(V) 终端(T) 转到(G) 帮助(H)
Uniform Multi-Platform E-IDE driver Revision: 6.31
ide: Assuming 50MHz system bus speed for PIO modes; override with idebus=xx
eth0: cs8900 rev K(3.3 Volts) found at 0xd0000300
cs89x0 media RJ-45, IRQ 37
SCSI subsystem driver Revision: 1.00
scsi0 : SCSI host adapter emulation for IDE ATAPI devices
UDA1341 audio driver initialized
NAND device: Manufacturer ID: 0xec, Chip ID: 0x76 (Samsung NAND 64MiB 3,3V)
NAND_ECC_NONE selected by board driver. This is not recommended !!
Creating 3 MTD partitions on "NAND 64MiB 3,3V":
0x00000000-0x00030000 : "boot"
0x00030000-0x00200000 : "kernel"
0x00200000-0x04000000 : "rootfs"
usb.c: registered new driver usbdevfs
usb.c: registered new driver hub
usb-ohci.c: USB OHCI at membase 0xe9000000, IRQ 26
usb.c: new USB bus registered, assigned bus number 1
hub.c: USB hub found
hub.c: 2 ports detected
usb.c: registered new driver usb_mouse

建立了 3 个 MTD 分区，分别用来存放 bootloader、kernel 和 rootfs。（分区图）

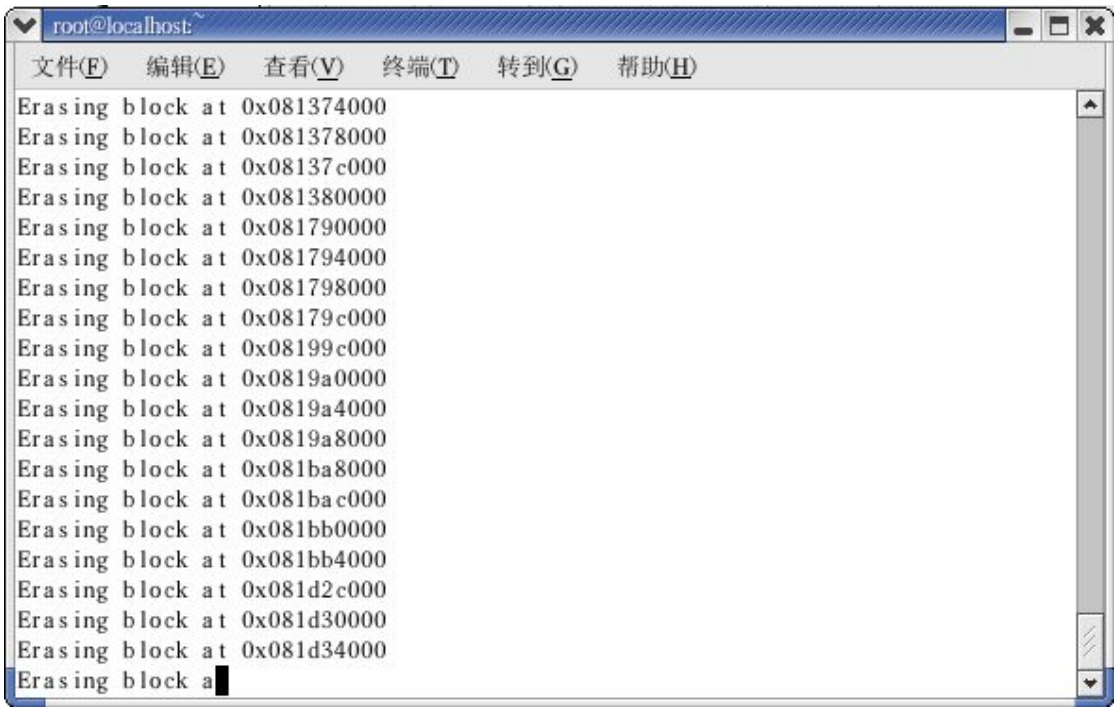
这里，我们通过运行 root/bin/下的 Writerootfs 脚本文件，可以一次性完成 rootfs 的下载。

在 minicom 中回车，输入：

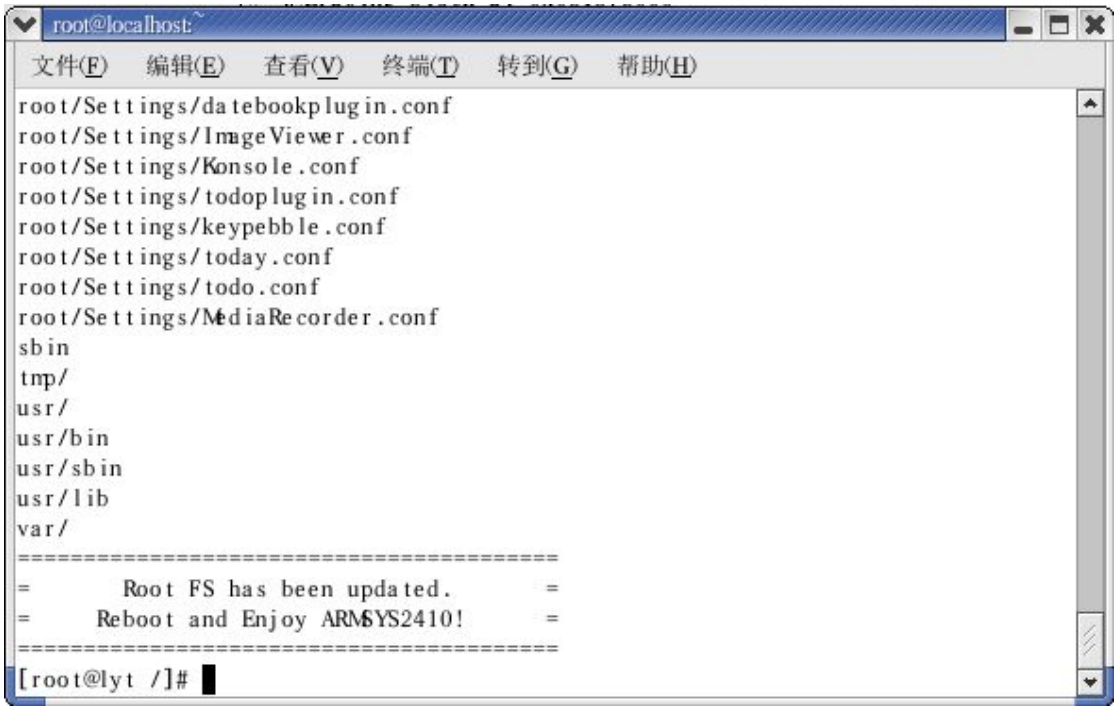
[root@lyt /]#cd /

[root@lyt /]#Writerootfs

显示结果如下：



上图中的过程是以 YAFFS 作为文件系统格式化分区。等待约 5 分钟，直到出现：



说明已经完成了 root 文件系统的更新，下面我们就可以从更新的文件系统启动了。

7．2 从 YAFFS 文件系统启动

按住 PC 键盘上的空格键，同时按下开发板的复位键。进入 vivi 命令行输入状态。重新设置系统启动参数，同样可以从 Linux\rootfilesystem\param_vivi 文本文件中拷贝获得：

param set linux_cmd_line=”noinitrd root=/dev/mtdblock/2 init=/linuxrc console=ttyS0”;

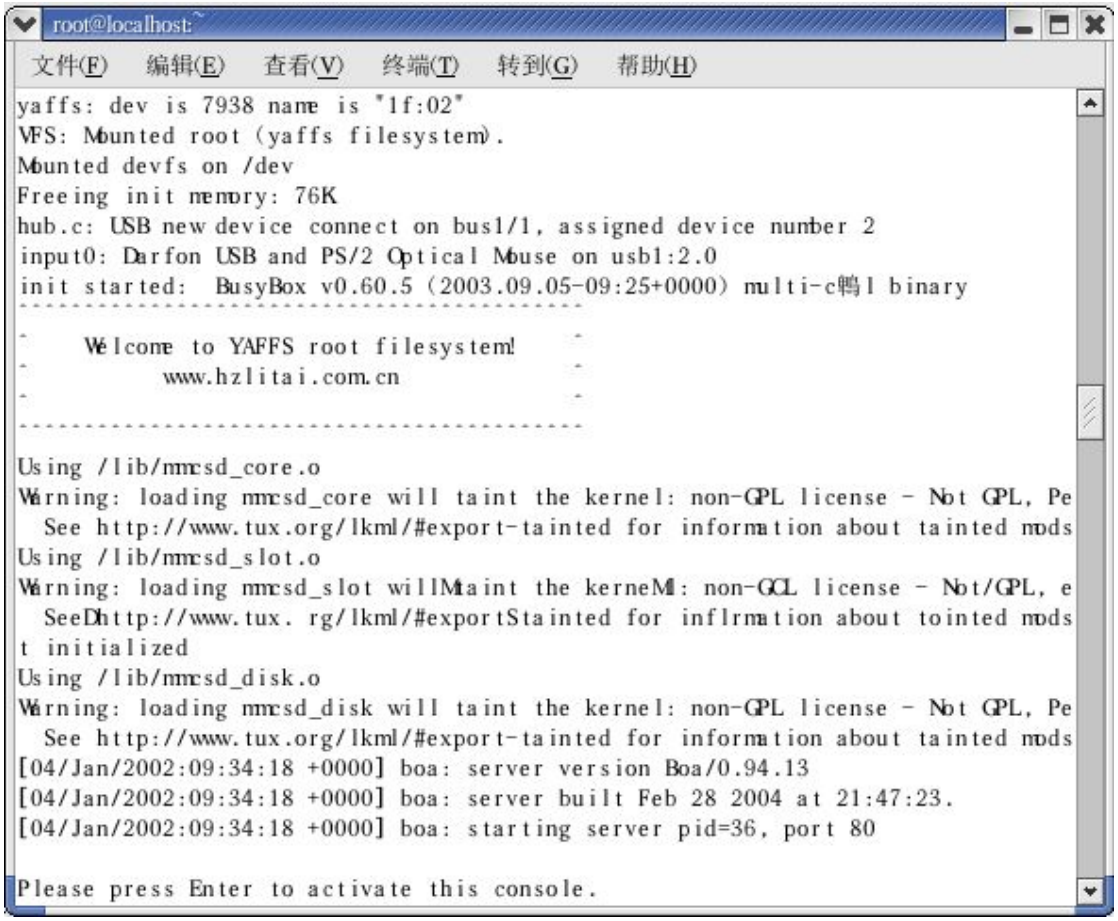
这里我们从 mtdblock/2 分区启动 root 文件系统。
可以用以下命令行保存该设置：

param save

为了验证能否成功启动文件系统，输入：

boot

启动信息如下：



以后，即使开发板掉电，root 文件系统仍旧保存。由于采用了可读写的 YAFFS 文件系统，你所保存的文件和设置在掉电后都不会丢失。

8. 主机上开发应用程序

ARMSYS2410 提供多个示范应用程序，它们以源码方式存放在光盘中 linux/examplesapp 下面。详细的介绍请参考本文的附录 C。这里我们仅就一个最简单的例子来说明基于 ARMSYS2410 开发应用程序的基本方法：

8．1 编写 Helloworld 代码

在/armsys2410/下建立 hello 目录：
cd /armsys2410
mkdir hello
编写以下代码，将它保存为 hello/hello.c

```
#include <stdio.h>
int main(void)
{
printf("Hello, ARMSYS2410.\n");
return 0;
}
```

8．2 编译 hello

输入以下命令交叉编译：
arm-linux-gcc –o hello hello.c

将在 hello 目录下生成 **hello** 可执行文件。

9．将应用程序下载到 ARMSYS2410

9．1 采用 NFS 方式

最为简单快捷的方法。按照第 6 节的介绍将目标板 mount 到主机，然后将 hello 拷贝到 root 文件系统的 bin 目录下，在目标板上运行 hello 即可。

9．2 采用 U 盘拷贝

把优盘插入 PC 的 USB 口，然后在终端执行以下命令把 hello 复制到优盘

```
mkdir /mnt/u
mount -t vfat /dev/sda1 /mnt/u
cp hello /mnt/u
umount /mnt/u
```

把优盘拔下来插入到 AMSYS2410 开发板的 USB HOST 端口，如果顺利，minicom 上将显示出系统识别出 U 盘的信息。

如果未能正确识别 U 盘，则有必要重新插拔 U 盘。

在 minicom 上输入以下命令行：

mkdir /tmp/u
mount -t vfat /dev/sda1 /tmp/u ； 挂接优盘
cd /tmp/u
./hello ； 执行 hello

9．3 采用 FTP 方式

采用 FTP 方式需要在 PC 机端建立一个 FTP 服务器。例如可以使用 Red Hat Linux9.0 中自带的 VSFTPD。通过 FTP 下载程序的主要步骤是：

把 hello 复制到 ftp 共享目录，然后在 minicom 上执行：

cd /bin ； 进入 bin 目录
ftp 192.168.253.1 ； 登录 ftp 服务器
>get hello ； 下载 hello
>bye ； 退出 ftp 登录
chmod a+x hello ； 改变 hello 的可执行权限
hello ； 执行 hello

10．QT 环境下进行设计开发

为了建立嵌入式 Qt 开发环境，光盘中 linux/Qt 目录下提供了以下资源：

- tmake 工具安装包：tmake-1.11.tar.gz

——用于生成应用工程的 Makefile 文件；

- Qt/Embedded 安装包：qt-embedded-2.3.7.tar.gz

——用于 Qt/Embedded 的安装；

- Qt 的 X11 版安装包：qt-x11-2.3.2.tar.gz

——用于产生一些必要的工具。

- Qtopia 安装包：qtopia-free-1.7.0.tar.gz

－－提供手持机的图形界面平台。

- 提供了两个脚本文件：buildarm 和 buildx86，用于在你的系统上安装基于 PC 的或者基于 ARM 的 Qt 开发环境。

10．1 基于 PC（主机）进行开发

10．1．1 建立基于 PC 的开发环境

（1） 将光盘中的资料拷贝到硬盘：

cp –av /dev/cdrom/linux/Qt /armsys2410/qt_x86

（2） 修改/etc/ld.so.conf 文件，修改后内容如下：

/armsys2410/qt_x86/qt/lib
/armsys2410/qt_x86/qtopia/lib
/usr/kerberos/lib
/usr/X11R6/lib
/usr/lib/sane
/usr/lib/mysql

（3）使用 buildx86 脚本完成一系列工具的安装和编译，只要输入：

cd /armsys2410/qt_x86
./buildx86

该过程时间比较长，需要约 30 分钟；

（4）输入：

ldconfig

运行 ldconfig 是为了使生成的 qt 和 qtopia 库有效，运行一次即可。

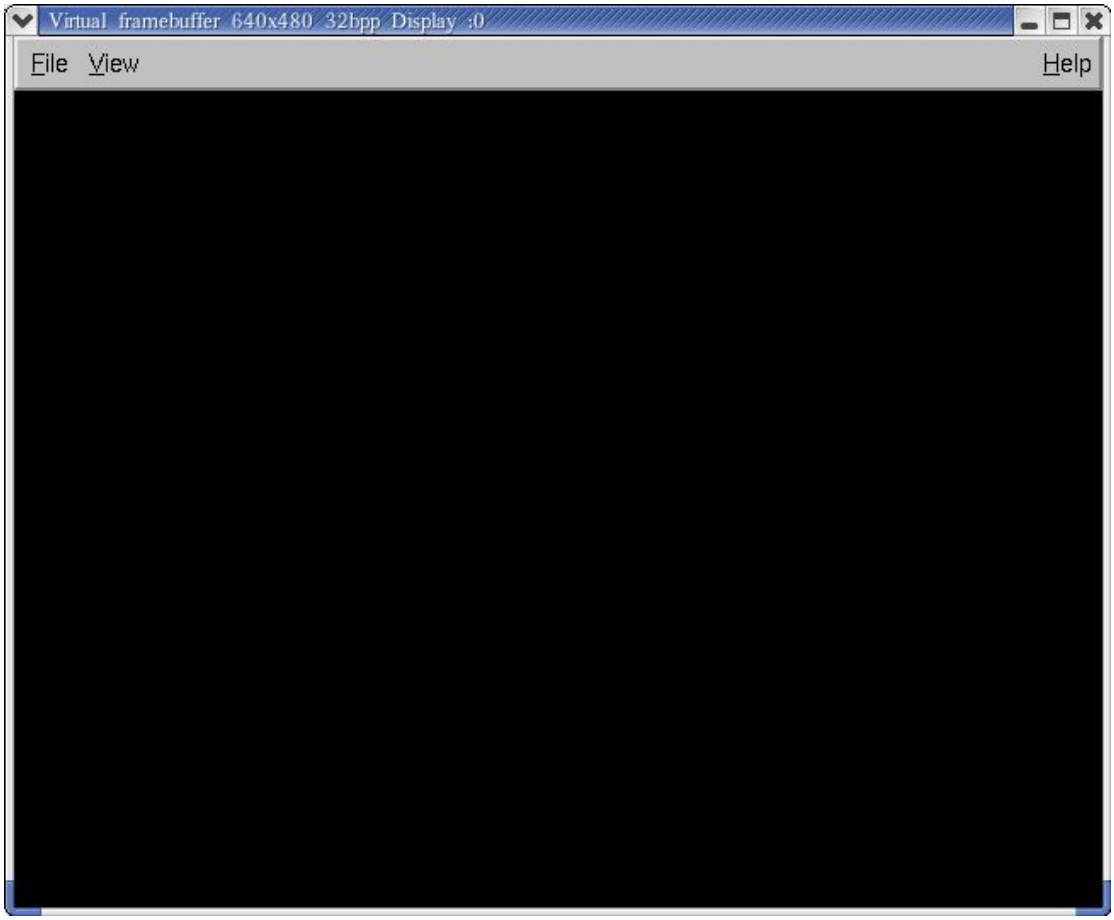
（5）在 PC 上模拟运行 Qtopia，如果液晶屏为 240×320，则输入：

qvfb& ；后台运行 qvfb

如果液晶屏为 640×480，则输入：

qvfb -width 640 -height 480 &

此时界面显示如下：



然后再运行：

qpe

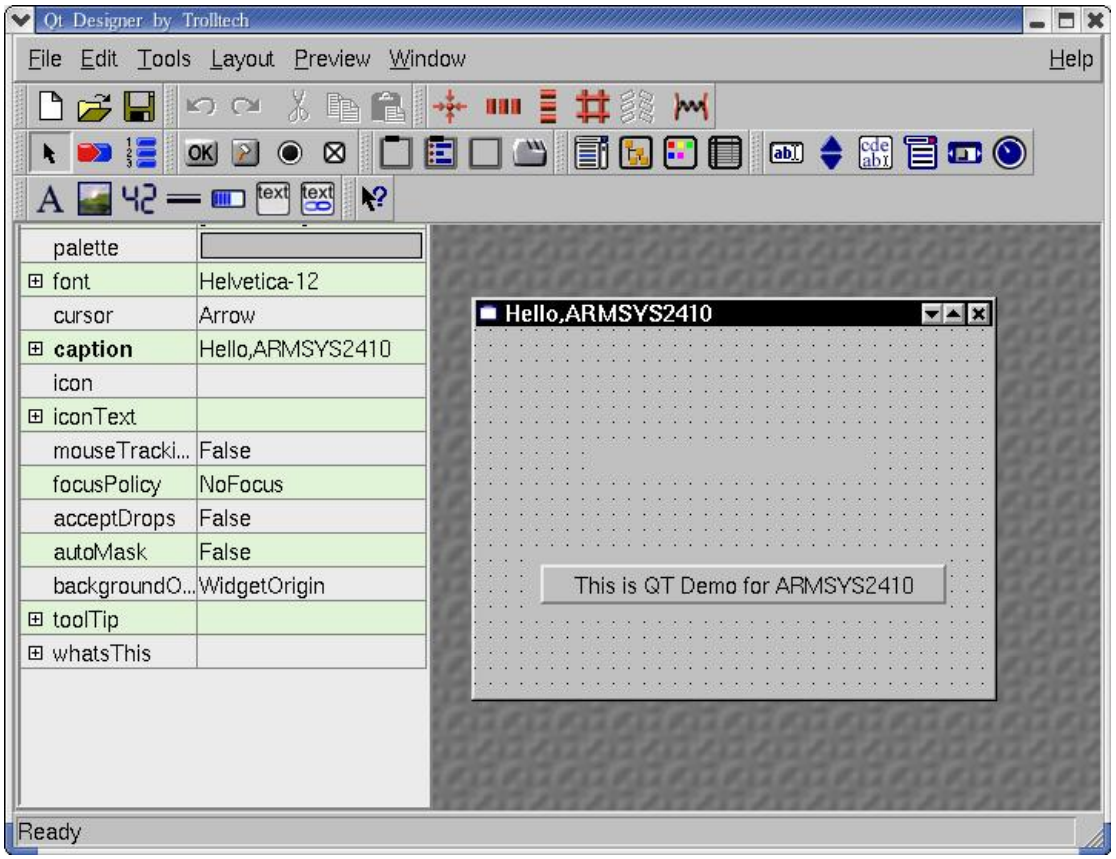
界面显示如下：

10．1．2 编写并编译应用程序

（1） 该目录下有一个名为 hello 的例子应用程序。

你也可以自己使用qt/bin 目录下提供的QT 设计器designer 来设计你的应用程序的界面，然后可以参照 hello 目录中提供的实例来制作其它必要的文件。

用 designer 来查看 Hello 的 ui 文件：



（2）编译之前首先设置环境变量：

cd /armsys2410/qt_x86
export QTDIR=$PWD/qt
export QPEDIR=$PWD/qtopia
export TMAKEDIR=$PWD/tmake
export TMAKEPATH=$TMAKEDIR/lib/qws/linux-generic-g++
export PATH=$QTDIR/bin:$QPEDIR/bin:$TMAKEDIR/bin:$PATH

这些输入保存在 setenv 文本文件中。

（3）将 hello 目录中的 makefile.x86 就地复制并更名为 Makefile；
然后开始编译 hello：

cd hello
make

编译完成后，将在/armsys2410/qt_x86/qtopia/bin 目录下将生成 hello 可执行文件。

10．1．3 制作图标

制作一个32 X 32 大小的PNG 格式的图标文件，将该文件存放在$QPEDIR/pics/inline

目录下，然后使用以下命令将$QPEDIR/pics/inline 目录下的所有图形文件转换成为一个c语言的头文件，这个头文件包含了该目录下的所有图形文件的rgb 信息。

```
qembed --images $QPEDIR/pics/inline/*.* > $QPEDIR/src/libraries/qtopia/inlinepics_p.h
```

这个输入保存在qembed文本文件中。

上面使用的qembed 工具是在qt for x11 上发布的工具。

10．1．4 制作桌面启动器（.desktop）文件

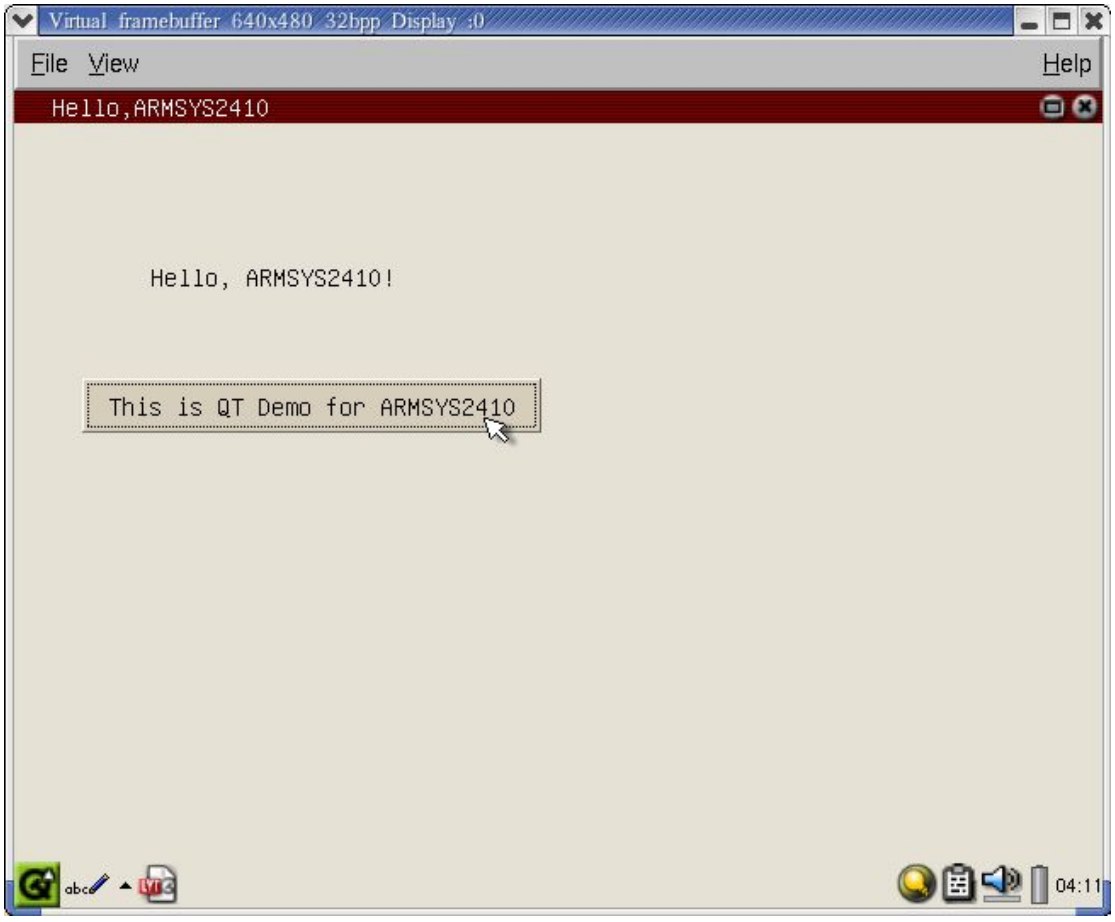
建立一个文本文件，在文件中添加以下的内容，这些内容指明了应用的名称，图标名等信息，然后将文件更名为xxxx.desktop,保存在$QPEDIR/apps/applications 目录下。例如对于hello，采用了以下启动器文件（hello.desktop）：

```
[Desktop Entry]
Comment=A Hello Program
Exec=hello
Icon=lyt_demo
Type=Application
Name=hello
```

10．1．5 运行应用程序

通过以下命令行在 qvfb 中单独运行 Hello：

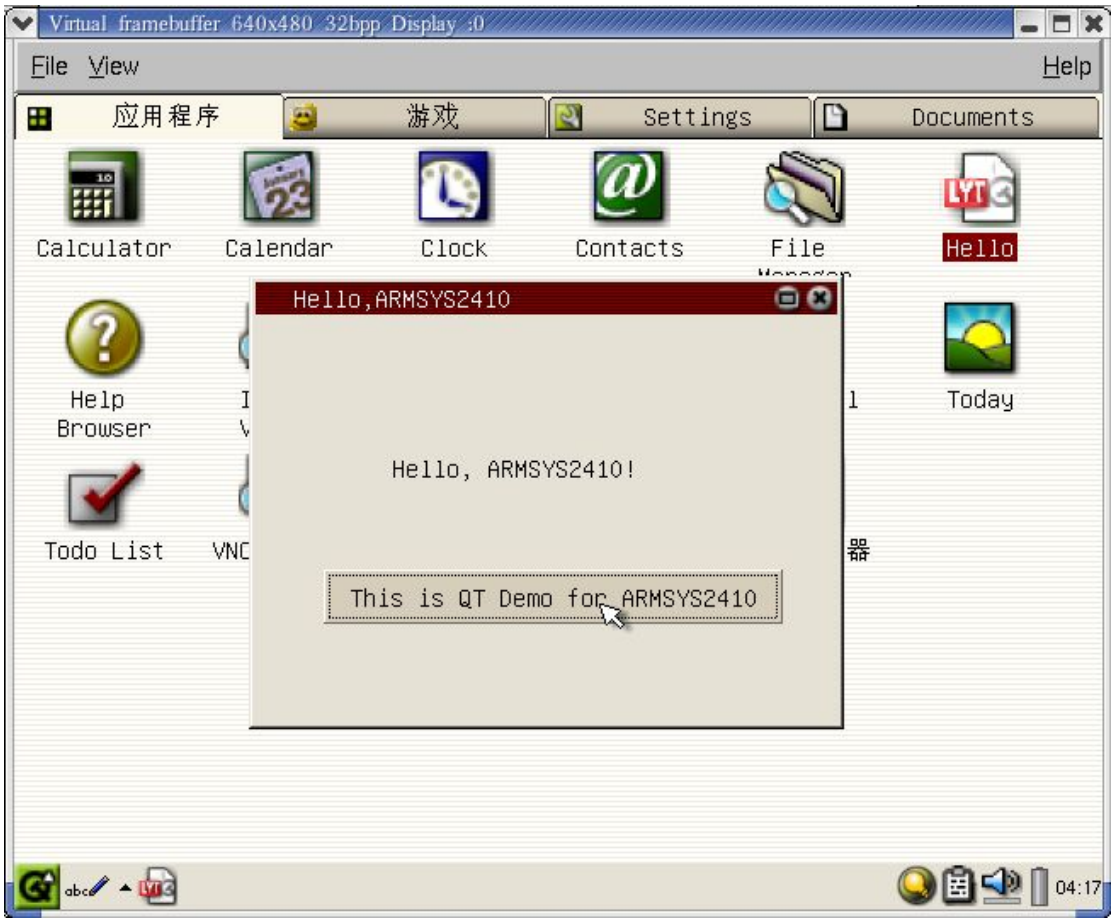
```
#qvfb -width 640 -height 480 &
#hello -qws
```



在 Qtopia 中运行 Hello

#qvfb -width 640 -height 480 &

#qpe



10．2 建立基于 ARM 的开发环境

（1）将光盘中的资料拷贝到硬盘：

cp –av /dev/cdrom/linux/Qt /armsys2410/qt_arm

（2） 安装开发环境

cd /armsys2410/qt_arm
./buildarm

10．2．1 编写并编译应用程序

同样可以使用该目录下的有一个名为 hello 的应用程序作为例子。
将 hello 目录中的 makefile.arm 就地复制更名为 Makefile；
编译 hello：

cd /armsys2410/hello
make

编译完成后，将在/armsys2410/qt_arm/qtopia/bin 目录下将生成 hello 可执行文件。

10．2．2 运行应用程序

参考第 9 节介绍的方式将 hello 下载到开发板的/opt/qtopia/bin 目录中。
在开发板上设置环境变量并运行 hello：
要在 ARMSYS2410 开发板上单独运行 hello，在 minicom 中输入以下命令行：

```
export set HOME=/root
export set QTDIR=/opt/qt
export set QPEDIR=/opt/qtopia
export set QWS_KEYBOARD="USB:/dev/input/event1"
export set QWS_MOUSE_PROTO="USB:/dev/input/mouse0"
export set PATH=$QPEDIR/bin:$PATH
export set LD_LIBRARY_PATH=$QTDIR/lib:$QPEDIR/lib
$QPEDIR/bin/hello -qws
```

要在 ARMSYS2410 开发板上单独运行 hello，在 minicom 中输入以下命令行：

```
export set HOME=/root
export set QTDIR=/opt/qt
export set QPEDIR=/opt/qtopia
export set QWS_KEYBOARD="USB:/dev/input/event1"
export set QWS_MOUSE_PROTO="USB:/dev/input/mouse0"
export set PATH=$QPEDIR/bin:$PATH
export set LD_LIBRARY_PATH=$QTDIR/lib:$QPEDIR/lib
$QPEDIR/bin/qpe > /dev/null 2>/dev/null
```

10．2．3 制作图标

制作一个32 X 32 大小的PNG 格式的图标文件，将该文件存放在$QPEDIR/pics/inline 目录下，然后使用以下命令将$QPEDIR/pics/inline 目录下的所有图形文件转换成为一个c 语言的头文件，这个头文件包含了该目录下的图形文件的rgb 信息。

```
qembed --images $QPEDIR/pics/inline/*.* > $QPEDIR/src/libraries/qtopia/inlinepics_p.h
```

注意上述的qembed 是在qt for x11 上发布的工具。

10．2．4 制作桌面启动器（.desktop）文件

建立一个文本文件，在文件中添加以下的内容，这些内容指明了应用的名称，图标名等信息，然后将文件更名为xxxx.desktop,保存在$QPEDIR/apps/applications 目录下。例如对于hello，采用了以下启动器文件（hello.desktop）：

```
[Desktop Entry]
Comment=A Hello Program
Exec=hello
Icon=lyt_demo
Type=Application
Name=hello
```

10．2．5 编译 QPE

重新编译 qpe：

cd ／armsys2410/qt_arm/qtopia/src
./configure –platform linux-arm-g++
make

10．3 重建根文件系统

在这里要利用原有的root_yaffs 的根文件系统映象，把新建的应用的相关文件添加到这个根文件系统中。这些文件包括：desktop文件，包含了图标的库文件libqte.so.＊ ，和应用程序的可执行文件。
具体过程如下：
（1） 拷贝desktop文件,在PC机终端上运行：

```
cp $QPEDIR/apps/Applications/hello.desktop
/armsys2410/root_yaffs/opt/qtopia/apps/Applications/

cp $QPEDIR/lib/libqpe.so.* /armsys2410/root_yaffs/opt/qtopia/lib/

cp $QPEDIR/bin/hello /armsys2410/root_yaffs/opt/qtopia/bin
```

将生成的新的根文件系统压缩后放到NFS文件系统root_nfs中，然后根据第6、7节的方式更新文件系统即可。

附录 A：内核驱动程序资源

ARMSYS2410 为以下设备提供完美地驱动程序支持，下面给出这些驱动程序的路径。

A.1 串口驱动

位于 kernel/drivers/serial/serial_s3c2410.c

A.2 MTD 驱动

位于 kernel/drivers/mtd/下；

A.3 ADC 驱动

位于 kernel/drivers/char/s3c2410-adc.c

A.4 4 按键驱动

位于 kernel/drivers/char/s3c2410_gpio_button.c

A.5 RTC 驱动

位于 kernel/drivers/char/s3c2410-rtc.c

A.6 触摸屏驱动

位于 kernel/drivers/char/s3c2410-ts.c

A.7 USB HOST 驱动

位于 kernel/drivers/usb/usb-ohci-s3c2410.c
kernel/drivers/usb/usb-ohci.o
kernel/drivers/usb/usbcore.c

A.8 CS8900A 网卡驱动

位于 kernel/drivers/net/cs8900a.c

A.9MMC 卡驱动

位于 kernel/drivers/mmcsd

A.10 USB 摄像头驱动

目前仅支持 ov511 芯片。
位于 kernel/drivers/usb/ov511.c

A.11UDA1341 声卡驱动

位于 kernel/drivers/sound/s3c2410-uda1341.c 支持录、放功能。

A.12 LCD 驱动

位于 kernel/drivers/video/s3c2410fb.c

A.12 LED 驱动

位于 kernel/drivers/char/example-leds.c

A.13 YAFFS 文件系统源码

位于 kernel/fs/yaffs，是可读写的文件系统。

A.14 其他资源

Kernel 中还提供了串行红外通讯、扩展标准键盘、扩展 PC 卡等。但目前 ARMSYS2410 没有提供这些硬件。

附录 B：实例程序

以下程序均是源码方式提供、除 Linux 系统之外的附加实例程序。如果要进行测试，进入 Linux 系统后，基于控制台操作即可。程序源码都位于/Linux/exampleapp 下。

B.1 LED 跑马灯程序

开机进入系统后，将会自动运行运行一个 LED 服务程序**(/etc/rc.d/init.d/leds)**，它其实是调用了 led-player 的一个脚本，led-player 开始运行后，将会在/tmp 目录下创建一个 **led-control** 管道文件，向该管道发送不同的参数可以改变 led 的闪烁模式：

#echo 0 0.2 > /tmp/led-control

运行该命令后，4 个用户 LED 将会以每个间隔 0.2 秒的时间运行跑马灯。

#echo 1 0.2 >/tmp/led-control

运行该后，4 个用户 LED 将会以间隔 0.2 秒的时间运行累加器。

B.2 LED 单独控制

led 是一个可以控制单个 led 的实用程序，要使用 leds 必须先停止 led-player，如下命令：

#/etc/rc.d/init.d/leds stop

该命令将停止 led-player 对 led 的操纵。led 的使用方法如下：

led

Usage: leds led_no 0|1

led_no 是要操作的 led(可为 0，1，2，3)，0 和 1 分别代表关闭和点亮。

#led 2 1

将点亮 LED3

B.3 按键控制

buttons 是一个简单的读取 4 个小按键的程序。

运行 buttons 后，当按下 K1 时，将打印“1”，松开时打印“129”；

运行 buttons 后，当按下 K2 时，将打印“2”，松开时打印“130”；

运行 buttons 后，当按下 K3 时，将打印“3”，松开时打印“131”；

运行 buttons 后，当按下 K4 时，将打印“4”，松开时打印“132”。

松开后的键值是按下的键值与“0x80”相或后得到的。

B.4 控制台下的 MP3 播放器

madplay 是一个基于控制台下的 mp3 播放器。它有多种播放控制模式，可以运行“madplay -h”查看其使用帮助，如下：

#madplay -h

播放例子音乐：

#madplay /Documents/1943.mp3

该命令将以缺省模式播放 1943.mp3 文件。

B.5 通过 USB 摄像头抓图 vidcat

vidcat 是一个基于字符界面的摄像头图象抓取程序，运行“vidcat -h”可以查看它的详细使用信息。

B.6 Web 服务器程序

Boa 程序实例。

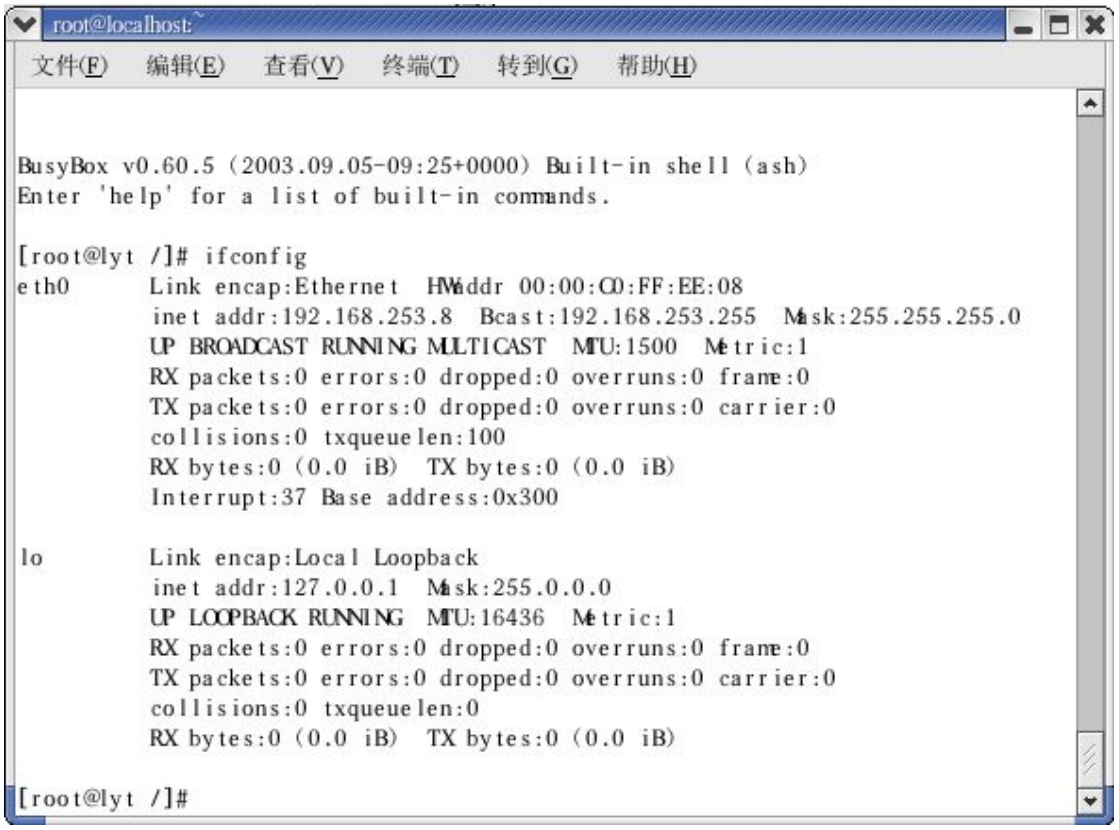
B.7 Qt 下的 Hello 程序

Qt 下的例子程序，展示了基本窗体和控件的使用，位于/linux/Qt 下。

附录 C：网络服务

C.1 ifconfig 命令

ifconfig 命令可检查并设置主机的网络接口，在 ARMSYS2410 上执行时不加任何参数会显示网卡与本地回路(lo，up，loopback)的信息。



运行不带参数的 **ifconig**，除了可查看网卡的状态外，还能改变一些网络的设置：

#ifconfig eth0 192.168.253.2 netmask 255.255.255.0

表示设置网卡 1 的地址 192.168.253.2，掩码为 255.255.255.0，不写 netmask 参数则默认为 255.255.255.0。ARMSYS2410 缺省网络 IP 地址为 192.168.253.2，这是通过在启动脚本 /etc/init.d/rcS 文件中使用 ifconfig 实现的。

也可以利用 ifconfig 命令暂时关闭网卡：

#ifconfig eth0 down

route 是 Linux 系统中控制台下用来设置网关的常用工具，它的使用方法如下：

#route

显示当前路由设置情况

#route add default gw 192.168.253.1

表示设置 192.168.253.1 为默认的路由

#route del default

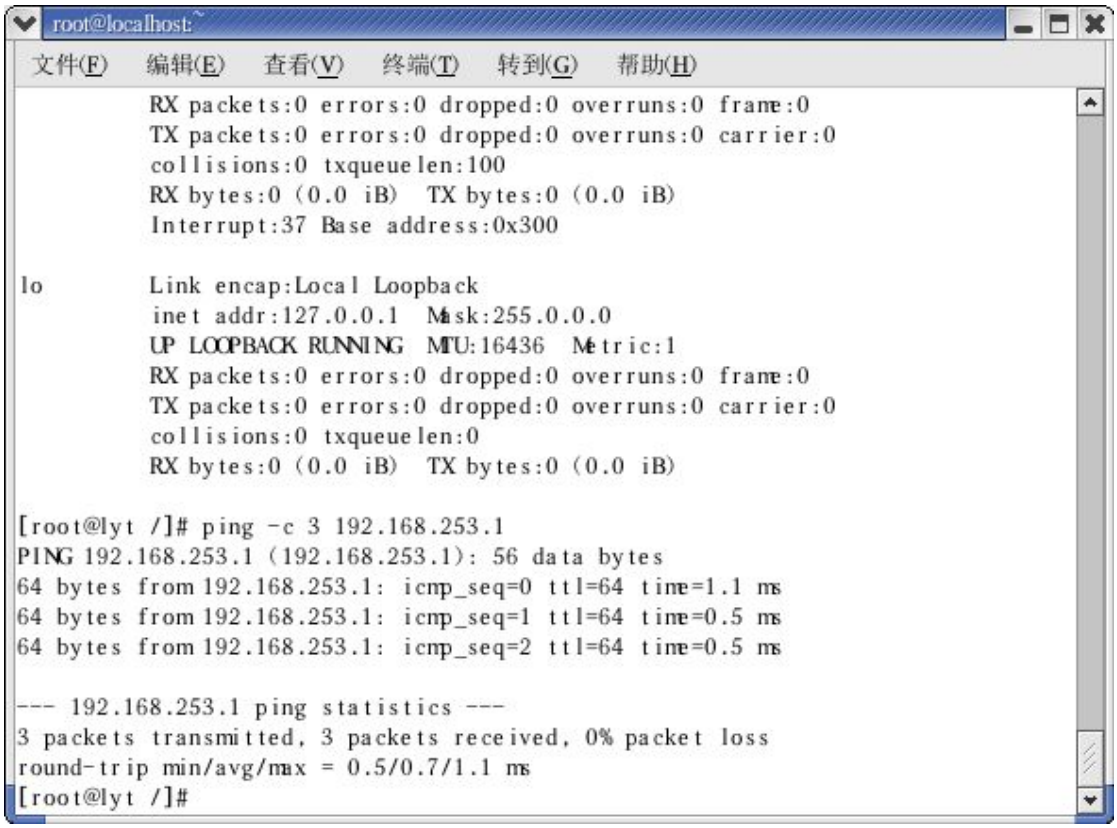
表示将默认的路由删除

C.2 ping 命令

ping 命令可以用来测试本机和网络上的另一台计算机是否连通。

ping –c 3 192.168.253.1

表示向 192.168.253.1 连续发送三次测试包，以验证网络是否连接正常，如果连接正常，则结果如下图所示：



C.3 traceroute 命令

traceroute 命令可用来检测本机连到其他主机时，数据包在传送过程中经过了哪些路由器及时间。

C.4 远程登陆 telnet

telnet 是一个经常被使用的远程登录工具，关于 telnet 的详细功能和用法可参考网络相关资料。

当 ARMSYS2410 所连接的局域网中，有一台机器提供了 telnet 服务，您就可以使用 ARMSYS2410 所带的 telnet 登录该机器了，当然，您必须有相应的帐号和密码。

假设提供了 telnet 服务的机器的 IP 地址为 192.168.253.1，使用以下命令登录该机器：
telnet 192.168.253.1

键入用户名和密码就可以远程使用该机器了。

使用 **telnet** 登录 **bbs**

当 ARMSYS2410 所连接的网络可以通到互联网，您可以配置 ARMSYS2410 登录外面的 bbs，使用 ifconfig 命令配置 ARMSYS2410 的 IP 地址，使用 route 命令配置网关：

route add default gw 192.168.253.1

C.5 Telnet 服务器 inetd

当 ARMSYS2410 已经被设置了 IP 地址后，它就可以作为一台 telnet 服务器了，这是因为系统启动的时候已经在后台执行了 inetd 服务程序。

ARMSYS2410 的缺省 IP 地址被设置为 192.168.253.8，则在“开始->运行”中敲入以下命令：

telnet 192.168.253.8

输入用户名 **root**，不需要密码按回车，即可登录 ARMSYS2410。

C.6 远程文件传送 ftp

ftp 的使用方法与标准 Linux 是相同的，假定您的 ftp 服务器 IP 地址是 192.168.253.1，在 shell 命令提示符下输入：

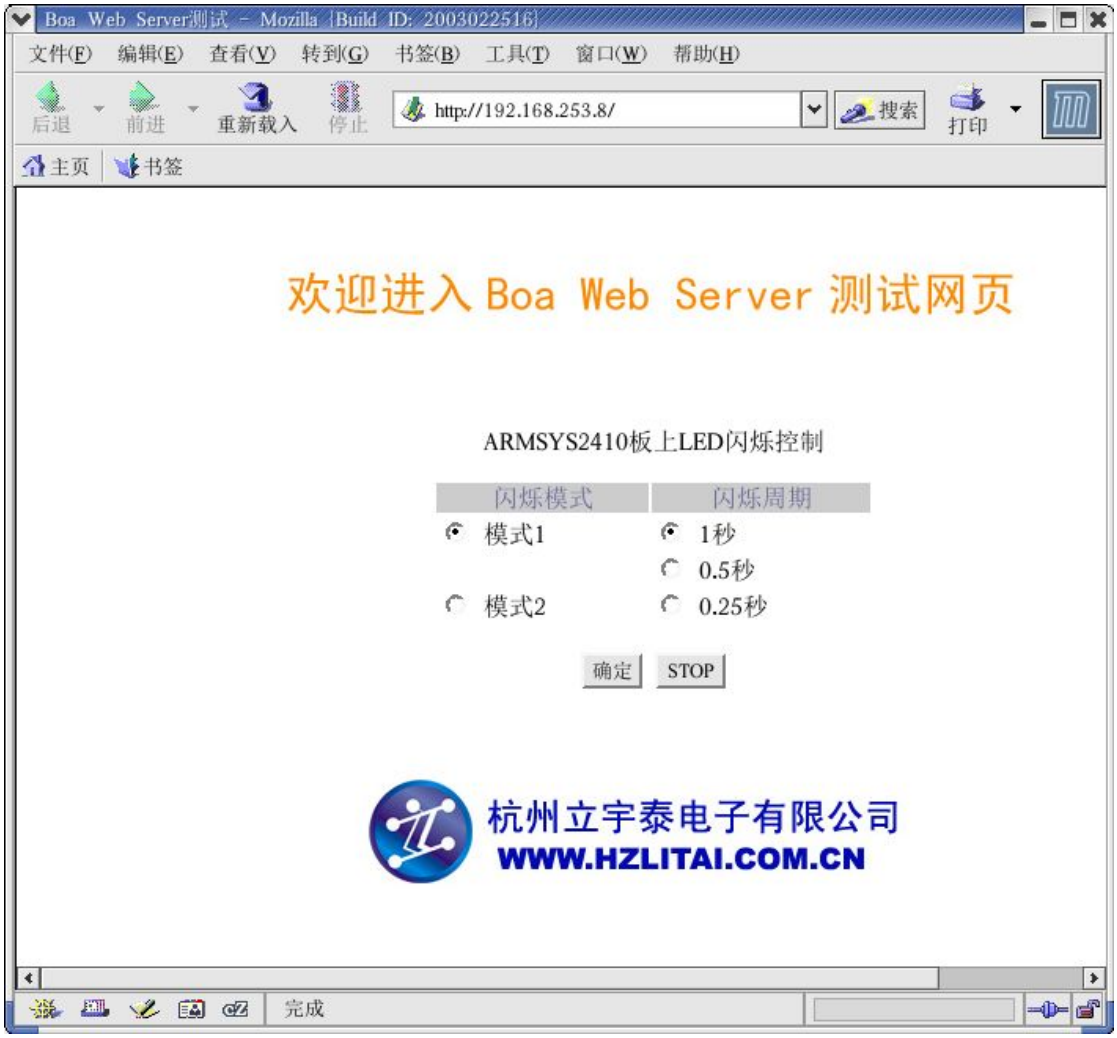
#ftp 192.168.253.1

C.7 Web 服务器 boa

系统开机后，并且 LED 计数器显示跳动正常，就可以使用 IE 或者其他浏览器来浏览存放于 ARMSYS2410 中的网页了，ARMSYS2410 的默认 IP 地址是 192.168.253.8，并且自动运行了 web 服务程序(boa)则在浏览器的地址栏中输入：

http://192.168.253.8

此时会跳出如下图所示页面，这表明您已经通过浏览器连接到 ARMSYS2410 了。



您可以使用网页中的各个测试项目进行测试，其中的按钮被按下后会通过 CGI 程序来控制板上的 LED 灯，包括 2 种闪烁模式和 3 种闪烁周期

如果要停止 web 服务器，则在 shell 命令提示符下输入以下命令，如下图所示：

#/etc/rc.d/init.d/httpd stop

要重新启动则输入：

#/etc/rc.d/init.d/httpd start